

dissecando jogos

Os sete jogos contidos neste volume tem uma característica única: o leitor participa do processo de criação acompanhando a montagem linha por linha.

Modificações e aperfeiçoamentos são propostos e resolvidos com detalhes. Ao terminar este livro o leitor além de ter sete jogos geniais, terá adquirido a habilidade de programar com criatividade.

Uma obra didática indispensável ao usuário de um micro da linha SINCLAIR: Ringo R-470, CP-200, TK 82/83/85, AS-1000, SINCLAIR ZX-81 ou TS-1000.

coleção
URANIA

CARLOS EDUARDO ROCHA SALVATO (DUDA) é um autor extremamente jovem (nasceu em 1968 e ainda está cursando o colegial), mas apesar disso é um veterano. Como autor de Coleção de Programas Vol. I (Editora Moderna 1983), foi um dos pioneiros a publicar material para micros da linha SINCLAIR. Posteriormente, adquiriu uma notável experiência didática ao colaborar com a revista Micro-hobby como responsável pela seção Dissecando.

Esta experiência está sendo transportada ao leitor nessa obra, onde o Duda diseca programas, mostrando o "como" e o "porquê" de cada linha.

Os programas são de jogos divertidos, sua análise é feita com a dedicação e a seriedade de um profissional, que já se tornaram a marca registrada dos trabalhos do Duda.



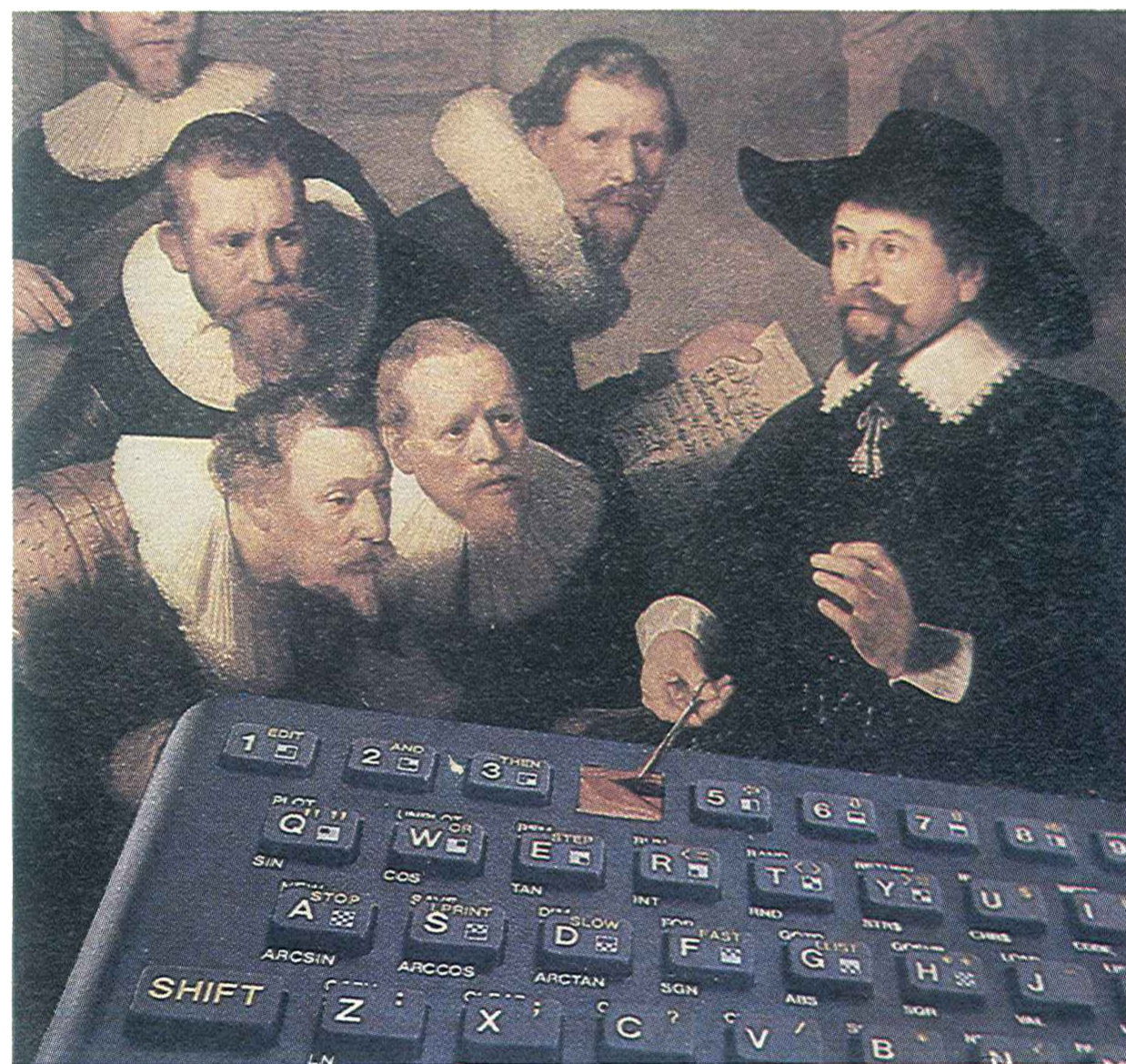
Xleph



DISSECANDO JOGOS

em basic comentado
linha por linha

carlos eduardo rocha salvato



para TK 82/83/85, CP 200
RINGO, SINCLAIR ZX-81



coleção
URANIA

dissecando jogos

coleção
URANIA

carlos eduardo rocha salvato

X
Neph



Este livro foi editado pela
ALEPH PUBLICAÇÕES E
ASSESSORIA PEDAGÓGICA LTDA.

Os programas foram digitados e testados no
NÚCLEO DE ORIENTAÇÃO DE ESTUDOS



Depto. de Cursos de Computação
Av. Brig. Faria Lima, 1451 - Conj. 31
01451 - São Paulo - SP
Tel.: (011) 813-4555
sob a coordenação da
prof. Betty Fromer Piazzi

CIP-Brasil. Catalogação-na-Publicação
Câmara Brasileira do Livro, SP

S173d Salvato, Carlos Eduardo Rocha, 1968-
Dissecando jogos / Carlos Eduardo Rocha Salvato. - São
Paulo : ALEPH: Ed. Moderna, 1985.

1. Jogos eletrônicos 2. Microcomputadores - Programação
3. Basic (Linguagem de programação para computadores) I.
Título.

	17. CDD-651.8
	18. -001.6425
	18. -001.642
85-0194	18. -001.6424

Índices para catálogo sistemático:

1. BASIC : Linguagem de programação : Computadores :
Processamento de dados 651.8 (17.) 001.6424 (18.)
2. Jogos : Computadores : Programas : Processamento
de dados 651.8 (17.) 001.6425 (18.)
3. Microcomputadores : Linguagem de Programação :
Processamento de dados 651.8 (17.) 001.642 (18.)

dissecando jogos

em basic TK comentado
linha por linha

para SINCLAIR ZX-81
RINGO, TK 82/83/85, CP 200


Aleph



Coordenação Editorial:

Pierluigi Piazzì

Avaliação, editoração, revisão técnica e produção gráfica:

Nancy Mitie Ariga

Ilustrações e Capa:

Piazzì, Moretti e Rembrandt

Foto de capa:

Hugo Sérgio Faleiros V.V.E.

Past-up e arte:

Fernando Moretti

Revisão e Copy-desk:

Lúcia Kairovsky

Produção:

Rosa Kogan Fromer

Todos os direitos reservados

Nleph

Publicações e Assessoria Pedagógica Ltda.

Av. Brig. Faria Lima,

1451 - Conj. 31

01451 - Jd. Paulista - São Paulo - SP

Tel.: (011) 813-4555



Distribuição exclusiva em livrarias

EDITORA MODERNA

Rua Afonso Brás, 431

04511 - São Paulo - SP

Tel.: (011) 531-5099

1985

Brasil

índice

Nota do editor	7
Capítulo 1 — Senha	9
Capítulo 2 — Porto	19
Capítulo 3 — Ambulância	31
Capítulo 4 — Escavações	43
Capítulo 5 — Tiro ao alvo	53
Capítulo 6 — Prédios	59
Capítulo 7 — Ataque submarino	69
Apêndice A	79
Apêndice B	81
Apêndice C	83
Apêndice D	85

nota do editor

Quando o "Duda" lançou seu primeiro livro (Coleção de Programas Vol. I, Editora Moderna 1983), a maior preocupação foi a de atender a um tipo particular de solicitação. Havia uma quantidade grande de usuários de computadores da linha SINCLAIR que só dispunham de software através de algumas poucas fitas e algumas raras publicações.

O momento agora é outro: o software disponível aumentou exponencialmente, bem como o número de publicações. O usuário tem farta escolha para alimentar seu computador.

Do ponto de vista didático, porém, não se fez muita coisa nestes dois anos. O usuário foi encarado como um simples consumidor e não como um possível futuro produtor.

Foi pensando nisso que planejamos este livro. Nele os programas contendo jogos em BASIC são fornecidos ao leitor com uma característica que os diferencia de todos os demais: eles são explicados e comentados linha por linha. São propostas alterações e aperfeiçoamentos, fazendo o leitor participar do processo de criação.

Para o aprendizado, esta técnica é de inestimável valor, sendo muito mais eficiente do que tratados e mais tratados de considerações teóricas.

"Unir o útil ao agradável" é um chavão muito batido, mas de uso inevitável para descrever a última obra do nosso mais jovem autor. A digitação dos programas produz resultados divertidos, mais do que isto, obtém-se no fim um sub-produto extremamente importante, e de forma suave e eficaz aprende-se a programar e criar.

Prof. Pierluigi Piazzì

1
senha



1 senha

INSTRUÇÕES

Este programa simula o conhecido jogo SENHA. Só que desta vez o computador esconderá o número. O número escolhido pelo computador é formado por 4 algarismos. Esses algarismos podem ser números de 0 a 5, e podem ser repetidos.

SENHA

Figura 1

```
0099  
7766  
5544 B
```

“ ”

Você deverá digitar um número de 4 algarismos, igual ao que o computador escolheu. Se o número que você digitou não for exatamente igual ao do computador, ele fornecerá os seguintes dados:

1 - Tendo algum algarismo igual ao número que o computador escolheu, mas fora do lugar, o dado será representado pela letra 'B'.

2 - Tendo algum algarismo igual ao número que o computador escolheu e na posição correta, o dado será representado pela letra 'P'.

Sempre que o computador apresentar um dado, apenas dirá o que foi especificado nos itens anteriores, não dirá qual é o número correspondente.

Você só terá 6 chances de acertar o número.

```

10 PRINT AT 0,12;"SENHA"
20 DIM N$(4)
30 DIM G$(4)
40 DIM W$(4)
50 LET C=0
50 RAND
70 FOR I=1 TO 4
80 LET N$(I)=CHR$ INT (RND*5+2
9)
90 NEXT I
100 PRINT AT 4,0;
110 LET C=C+1
120 IF C=7 THEN GOTO 360
130 INPUT G$
140 IF G$="" THEN GOTO 360
150 PRINT TAB 12;G$;" "
160 IF G$=N$ THEN GOTO 350
170 LET W$=N$
180 FOR I=1 TO 4
190 IF G$(I) <> W$(I) THEN GOTO 2
30
200 LET W$(I)="P"
210 PRINT "P";
220 LET G$(I)="/"
230 NEXT I
240 FOR I=1 TO 4
250 IF W$(I)="P" THEN GOTO 320
260 FOR J=1 TO 4
270 IF I=J OR W$(I) <> G$(J) THEN
GOTO 310
280 LET G$(J)="?";
290 PRINT "B";
300 GOTO 320
310 NEXT J
320 NEXT I
330 PRINT
340 GOTO 110
350 PRINT "PARABENS"
360 PRINT "O NUMERO ERA ";N$
370 PRINT AT 21,0;"DESEJA JOGAR
NOUAMENTE ? (S/N)"
380 IF INKEY$="S" THEN GOTO 410
390 IF INKEY$="N" THEN GOTO 440
400 GOTO 370
410 SLOW
420 CLS
430 GOTO 10
440 CLS
450 PRINT AT 10,13;"F I M"
460 STOP
470 SAVE "SENHA"
480 RUN
490 STOP

```

```

SENHA
3443 PPP
3473 PPP
3413 PARABENS
O NUMERO ERA 3413

```

Figura 2

```
DESEJA JOGAR NOUAMENTE ? (S/N)
```

EXPLICAÇÃO

1ª Parte – Iniciação (linha 10 a 60)

linha 10 - Imprime a palavra SENHA na posição (0,12) do vídeo.

linha 20

a 40 - Essas linhas dimensionam as variáveis N\$, G\$, W\$ respectivamente. Como o número que aparece entre parênteses nestas linhas é 4, isto quer dizer que estas variáveis sempre terão 4 dígitos de comprimento.

linha 50 - Zera a variável que contará o número de chances que você tem para acertar o número que o computador escondeu.

linha 60 - Cria uma nova semente, que irá gerar os números aleatórios.

2ª Parte – Esta parte gera o número que o computador escolherá. (linha 70 a 90)

linha 70 - Inicia um LOOP de quatro voltas.

linha 80 - A variável N\$ guardará o número que o computador está escondendo. (Veja mais detalhes sobre esta linha na parte de MODIFICAÇÕES).

linha 90 - Finaliza o LOOP.

3ª Parte — início do vídeo (linha 100)

linha 100 - Desloca o início do vídeo para a quarta linha.

4ª Parte — Verificação do fim do jogo (linha 110 a 120)

linha 110 - Incrementa a variável contadora das chances.

linha 120 - Verifica se a variável contadora é igual ao número de chances mais um, se esta condição for verdadeira o programa salta para linha 360, caso contrário continua o programa.

5ª Parte — Entrada e impressão do número (linha 130 a 150)

linha 130 - Espera que você entre com um número de quatro algarismos. Este número que você digitou será armazenado na variável G\$.

linha 140 - Se você não entrou com nenhum número o programa vai para linha 360, se você entrou com um número o programa continua.

linha 150 - Imprime na décima segunda coluna o número que você digitou seguido de um espaço.

6ª Parte — Colocação do 'P' (linha 160 a 230)

linha 160 - Verifica se o número que você digitou é igual ao do computador, se for o programa vai para linha 350, caso contrário, o programa continua.

linha 170 - Faz a variável W\$ ser igual ao número que o computador escolheu. Isto é feito para que durante a comparação entre os números não se perca o número do computador.

linha 180 - Inicia um LOOP de quatro voltas.

linha 190 - Esta linha irá comparar o número que você digitou com o número do computador, se os números forem iguais o programa continua, caso contrário vai para a linha 230.

linha 200 - Faz a variável W\$ ser igual à letra 'P'. Isto é feito para que o computador coloque a letra 'P' no vídeo e evite fazer a comparação para colocar o 'B'.

linha 210 - Imprime a letra 'P' no vídeo.

linha 220 - Faz a variável G\$ ser igual a '/'. Isto é feito caso o número escolhido pelo computador tenha algarismos repetidos.

linha 230 - Finaliza o LOOP.

7ª Parte — Colocação do 'B' (linha 240 a 320)

linha 240 - Inicia um LOOP de quatro voltas.

linha 250 - Se a variável W\$ for igual à letra 'P', o programa vai para linha 320, caso contrário o programa continua. Esta condição é necessária senão o número que está na posição correta, terá uma letra 'P' e uma letra 'B' para representá-lo.

linha 260 - Inicia um "sub-LOOP" de quatro voltas.

linha 270 - Nesta linha são estabelecidas duas comparações. A primeira compara as variáveis I e J, se elas forem iguais o programa vai para a linha 310, isto acontece por uma razão muito óbvia: se as variáveis I e J forem iguais, isso quer dizer que o computador está comparando dois números na mesma posição, se os números forem iguais já deverá ter sido atribuída a letra 'P', caso contrário não seria atribuído nenhum código. Outra comparação que também é feita nesta linha é a comparação de um número do computador com todos os seus números, caso exista um número igual mas fora da posição o programa continua, mas se não existir nenhum número igual o programa vai para linha 310.

linha 280 - Se no seu número existir um número igual ao do computador, mas ele estiver fora da posição a variável G\$ fica sendo igual a "?".

linha 290 - Imprime a letra 'B' no vídeo.

linha 300 - Manda o programa para linha 320, senão os números repetidos também serão computados.

linha 310 - Finaliza o "sub-LOOP".

8ª Parte — Congratulações (linha 350)

linha 350 - Imprime a palavra "PARABÉNS" no vídeo.

9ª Parte — Impressão do número correto (linha 360)

linha 360 - Imprimirá no vídeo o número correto.

10ª Parte — Finalização (linha 370 a 460)

Obs.: Esta parte será usada em quase todos os programas deste livro. Para evitar repetições nos outros programas, deixarei apenas indicado.

linha 370 - Imprime na posição (21,0), se você quer jogar de novo.

linha 380 - Se você pressionar a letra 'S' o programa irá para linha 410.
 linha 390 - Se você pressionar a letra 'N' o programa irá para linha 440.
 linha 400 - Se você não pressionar nenhuma dessas duas letras o programa volta para linha 370.
 linha 410 - Põe o computador no modo SLOW.
 linha 420 - Limpa todo o conteúdo do vídeo.
 linha 430 - Manda o programa para linha 10.
 linha 440 - Limpa todo o conteúdo do vídeo.
 linha 450 - Imprime na posição (10,13) a palavra "FIM" em vídeo inverso.
 linha 460 - Pára o programa.

11ª Parte — Auto-Start do programa (linha 470 a 490)

linha 470 - Armazena em fita o programa com nome "SENHA".
 linha 480 - Roda o programa.
 linha 490 - Pára o programa.

```

SENHA

1122 P
1133 P
1444 B

4144 P
1477 B

4148 P
O NUMERO ERA 0100

```

Figura 3

```

DESEJA JOGAR NOVAMENTE ? (S/N)

```

MODIFICAÇÕES

Modificação 1

Uma modificação muito simples de ser feita, é a da linha 80. Esta linha define qual o número que será escolhido pelo computador. No jogo normal o computador só escolhe números entre 0 e 4. Para modificá-lo deve-se alte-

rar o número que vem logo após a instrução RND. No programa este número é 5. Se você mudar este número para 8 o computador escolherá números entre 0 e 7 aumentando a dificuldade do jogo. Você também pode diminuir a dificuldade do jogo diminuindo este número. Se você colocar um número maior que 10 o computador também poderá escolher letras e números ao invés de escolher só números.

Modificação 2

Outra modificação que pode ser facilmente feita é a alteração do número de chances que você tem para acertar o número que o computador escolheu. Isso pode ser feito trocando na linha 120 o número que aparece logo após o sinal de igual e que deve ser igual ao número de chances que você quer ter, mais um.

Modificação 3

Essa modificação será um pouco mais complicada, porque atingirá várias linhas. Esta modificação alterará o tamanho do número que o computador escolherá.

Façamos de conta que você queira alterar o tamanho do número que o computador escolherá para um tamanho X. Você deverá alterar o valor de X, que no programa original é igual a 4, nas seguintes linhas: 20, 30, 40, 70, 180, 240, 260 e também deverá por na linha 140 X espaços-entre-aspas.

Modificação 4

Esta última modificação é extremamente complexa. Iremos acrescentar algumas linhas ao programa para que o número escolhido pelo computador não tenha algarismo repetidos.

Acrescente as seguintes linhas:

```

81 FOR J=1 TO 4
82 IF N$(I)=N$(J) AND I<>J THE
N GOTO 80
83 NEXT J

```

O computador demorará um pouco mais para escolher o número, tenha paciência ou se preferir acrescente mais estas duas linhas:

```

85 FAST
95 SLOW

```

Agora tente modificar o programa de modo que as alterações citadas anteriormente já sejam perguntadas pelo próprio programa. O Display inicial deverá ser o seguinte:

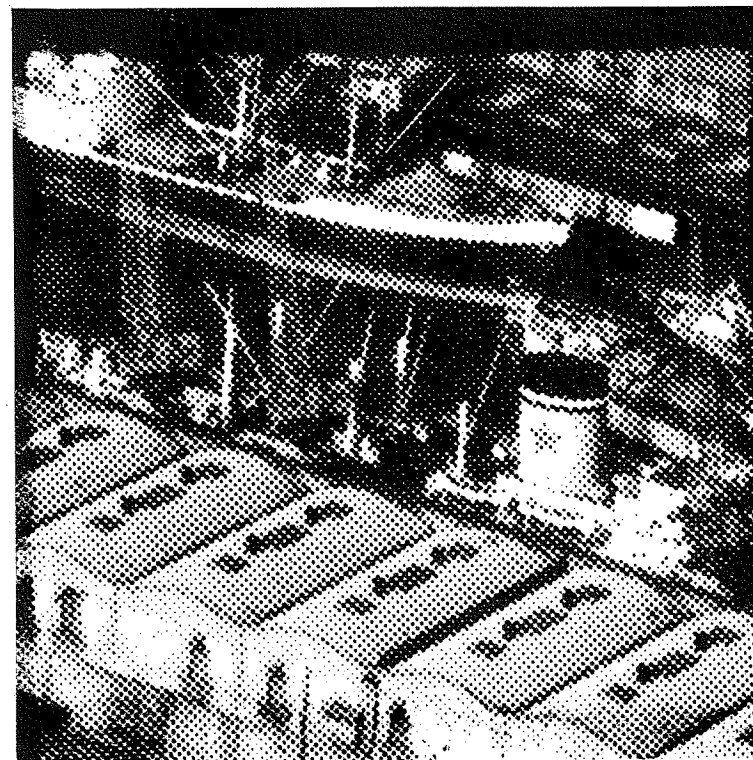
```
                SENHA
VOCE QUER NUMEROS REPETIDOS
N
QUANTOS ALGARISMOS TERA O NUMERO
QUE EU VOU ESCOLHER ?
5
EU POSSO ESCOLHER NUMEROS ENTRE
0 E ?
8
QUANTAS CHANCES VOCE QUER TER
PARA ACERTAR O NUMERO ?
ESTA BEM, AGUARDE ENQUANTO EU
ESCOLHO O NUMERO.
```

Após você ter modificado para que ele fique desse jeito, olhe o apêndice A e compare com a minha resposta.

QUADRO DE VARIÁVEIS PRINCIPAIS DO PROGRAMA

- N\$ - senha escolhida pelo computador
- G\$ - senha que você vai "chutar"
- W\$ - verifica se você acertou a senha
- C - número de "chutes" dado por você
- I - variável contadora na comparação das senhas
- J - variável contadora na comparação das senhas

2
porto



2
porto

INSTRUÇÕES

Logo após você ter digitado RUN você deverá ver na tela uma imagem assim:



Figura 1



No alto da tela você verá o número 3 que indica o número de chances que você tem. No centro está o porto onde você deverá descarregar a sua carga na segunda fase do jogo e o número 0 que indica quantos pontos você já fez.

Logo abaixo do número 3 existe um ponto que representa o seu navio, mas ele não estará obrigatoriamente colocado nesta posição, porque sua localização na tela é aleatória.

Na parte de baixo da tela estão localizados os três portos onde você buscará a sua carga, na primeira fase.

Logo após você digitar RUN o seu barco já sairá em direção aos três portos, e você deverá controlá-los através dos seguintes comandos:

- 5 - para a esquerda;
- 8 - para a direita;
- ou através do joystick.

Você precisa atracar exatamente no ponto mais saliente dos portos, caso contrário você baterá e perderá uma chance.

Se você atracar perfeitamente em um dos portos, a segunda fase se iniciará imediatamente.

A segunda fase é semelhante à primeira, só que você irá partir de um ponto aleatório na parte de baixo da tela em direção ao único porto na parte de cima da tela, para descarregar a sua carga.

Conforme você for atracando nos portos, na parte inferior da tela, eles vão diminuindo de tamanho mostrando que a carga existente nestes portos está acabando. Quando você conseguir acabar com a carga dos três portos, novas cargas serão colocadas.

```

10 LET T=3
20 LET X=0
30 LET B=4
40 LET C=B
50 LET D=B
60 LET F#=""
70 LET B#=""
80 LET C#=B#
90 LET D#=B#
100 CLS
110 IF B=0 AND C=0 AND D=0 AND
T>0 THEN GOTO 30
120 PRINT T;TAB 14;";";TAB 25;
X
130 PRINT AT 1,15;F#
140 PRINT AT 20,0;F#
150 PRINT AT 20,20;F#
160 PRINT AT 20,20;F#
170 PRINT AT 21,0;B#( TO B)
180 PRINT AT 21,10;C#( TO C)
190 PRINT AT 21,20;D#( TO D)
200 LET E=INT (RAND*50)+1
210 FOR Z=39 TO 3 STEP -1
220 IF INKEY#="S" THEN LET E=E-
1
230 IF INKEY#="8" THEN LET E=E+
1

```

```

240 PLOT E,Z
250 UNPLOT E,Z
260 NEXT Z
270 IF E=6 THEN LET B=B-1
280 IF E=26 THEN LET C=C-1
290 IF E=46 THEN LET D=D-1
300 IF B=-1 OR C=-1 OR D=-1 THE
N GOTO 440
310 IF E=6 OR E=26 OR E=46 THEN
GOTO 330
320 GOTO 470
330 LET E=INT (RAND*50)+1
340 FOR Z=3 TO 39
350 IF INKEY#="S" THEN LET E=E-
1
360 IF INKEY#="8" THEN LET E=E+
1
370 PLOT E,Z
380 UNPLOT E,Z
390 NEXT Z
400 IF E=30 THEN GOTO 420
410 GOTO 470
420 LET X=X+1
430 GOTO 100
440 IF B=-1 THEN LET B=0
450 IF C=-1 THEN LET C=0
460 IF D=-1 THEN LET D=0
470 PRINT AT 10,10;"SPLASH"
480 LET T=T-1
490 FOR Z=1 TO 50
500 NEXT Z
510 IF T=0 THEN CLS
520 IF T=0 THEN PRINT AT 10,5;
" F I M D E J O G O ";AT 12,10;
"PLACAR=";X
530 IF T=0 THEN GOTO 550
540 GOTO 100
550 PRINT AT 21,0;"DESEJA JOGAR
NOVAMENTE ? (S/N)"
560 IF INKEY#="S" THEN GOTO 590
570 IF INKEY#="N" THEN GOTO 520
580 GOTO 560
590 CLS
600 SLOW
610 GOTO 10
620 CLS
630 PRINT AT 10,10;" F I M "
640 STOP
650 SAVE "PORT0"
660 RUN
670 STOP

```



Figura 2

■



EXPLICAÇÃO

1ª Parte — Iniciação (linha 10 a 100)

- linha 10 - A variável T assume o valor 3. A variável T guardará o número de chances que você tem.
- linha 20 - Zera a variável X, que contará os seus pontos.
- linha 30 - A variável B assume o valor 4. A variável B definirá o tamanho do primeiro porto da parte inferior da tela.
- linha 40 - A variável C assume o valor igual ao de B. A variável C definirá o tamanho do segundo porto da parte inferior da tela.
- linha 50 - A variável D assume o valor igual ao de B. A variável D definirá o tamanho do terceiro porto da parte inferior da tela.
- linha 60 - O conteúdo da variável alfanumérica F\$ será o caractere número 4.
- linha 70 - O conteúdo da variável alfanumérica B\$ será igual a 4 espaços inversos.
- linha 80 - O conteúdo da variável alfanumérica C\$ será o mesmo da variável B\$.

linha 90 - O conteúdo da variável alfanumérica D\$ será o mesmo da variável B\$.

linha 100 - Limpa todo o conteúdo da tela.

2ª Parte — Checagem das cargas (linha 110)

linha 110 - Verifica se a carga dos três portos é igual a zero e também se você ainda tem chances de continuar o jogo. Se os portos não tiverem mais cargas, o programa vai para a linha 30 onde as cargas serão reestabelecidas, caso contrário o programa continua.

3ª Parte — Impressão dos Portos (linha 120 a 190)

linha 120 - Imprime na posição (0,0) a variável T, imprime na posição (14,0) os caracteres 133 e 128. Esta linha também imprime na posição (25,0) a variável X.

linha 130 - Imprime o conteúdo da variável F\$ na posição (15,1).

linha 140 - Imprime o conteúdo da variável F\$ na posição (3,20).

linha 150 - Imprime o conteúdo da variável F\$ na posição (13,20).

linha 160 - Imprime o conteúdo da variável F\$ na posição (23,20).

linha 170 - Imprime o conteúdo da variável B\$ até B na posição (3,21).

Exemplo: Se B for igual a 2 só serão impressos dois espaços inversos.

linha 180 - Imprime o conteúdo da variável C\$ até C na posição (21,13).

linha 190 - Imprime o conteúdo da variável D\$ até D na posição (21,23).

4ª Parte — Primeira Fase (linha 200 a 260)

linha 200 - Escolhe a posição inicial da qual sairá o seu navio, e coloca na variável E.

linha 210 - Inicia um LOOP que contará regressivamente de 39 até 3. Este LOOP serve para ser feita a descida até os portos.

linha 220 - Caso você pressione a tecla '5' o seu navio se deslocará de uma posição para a esquerda.

linha 230 - Caso você pressione a tecla '8' o seu navio se deslocará de uma posição para a direita.

linha 240 - Imprime um ponto (PLOT) na posição (E,Z).

linha 250 - Apaga o ponto que foi impresso na linha anterior. Isto é feito para que o ponto fique piscando.

linha 260 - Finaliza o LOOP.

5ª Parte — Checa se o navio bateu (linha 270 a 320)

- linha 270 - Se a variável E for igual a 6, quer dizer que você atracou no primeiro porto, por isso a variável B será diminuída de um, pois você já pegou uma carga lá existente, caso contrário o programa continua.
- linha 280 - Se a variável E for igual a 26, quer dizer que você atracou no segundo porto, por isso a variável C será diminuída de um, pois você já pegou uma carga lá existente, caso contrário o programa continua.
- linha 290 - Se a variável E for igual a 46, quer dizer que você atracou no terceiro porto, por isso a variável C será diminuída de um, pois você já pegou uma carga lá existente, caso contrário o programa continua.
- linha 300 - Checa se no porto que você atracou existe carga, se positivo o programa continua, caso contrário o programa vai para a linha 440.
- linha 310 - Checa se você atracou em um dos três portos, se atracou o programa vai para a segunda fase (linha 330).
- linha 320 - Manda o programa para linha 470.

6ª Parte — Segunda Fase (linha 330 a 390)

- linha 330 - Escolhe a posição inicial de onde sairá o seu navio e coloca na variável E.
- linha 340 - Inicia um LOOP que contará de 3 até 39. Este LOOP serve para que seja feita a subida até o porto.
- linha 350 - Caso você pressione a tecla '5' o seu navio se deslocará uma posição para a esquerda.
- linha 360 - Caso você pressione a tecla '8' o seu navio se deslocará uma posição para a direita.
- linha 370 - Imprime um ponto (PLOT) na posição (E,Z).
- linha 380 - Apaga o ponto que foi impresso na linha anterior, isto é feito para que o ponto fique piscando.
- linha 390 - Finaliza o LOOP.

7ª Parte - Checa se o navio bateu (linha 400 e 410)

- linha 400 - Se a variável E for igual a 30, significa que você atracou perfeita-

mente no porto superior e o programa vai para a linha 420, caso contrário o programa continua.

- linha 410 - Manda o programa para a linha 470.

8ª Parte — Soma de Ponto (linha 420 e 430)

- linha 420 - Soma um ponto a variável X, que no caso é o seu placar.
- linha 430 - Manda o programa para a linha 100 e reinicia a primeira fase.

9ª Parte — Batida do Navio (linha 440 a 500)

- linha 440 - Se você atracar no primeiro porto e houver carga, a variável B que guarda o número de cargas ficou sendo negativa e precisa ser zerada.
- linha 450 - Se você atracou no segundo porto e não havia carga, a variável C que guarda o número de cargas ficou sendo negativa e precisa ser zerada.
- linha 460 - Se você atracou no terceiro porto e não havia carga, a variável D que guarda o número de cargas ficou sendo negativa e precisa ser zerada.
- linha 470 - Imprime a palavra 'SPLASH' na posição (10,10).
- linha 480 - Diminua a variável T de um, diminuindo, portanto, as suas chances.
- linha 490 - Inicia um LOOP que conta de 1 até 50. Este LOOP serve apenas para dar uma pausa.
- linha 500 - Finaliza o LOOP.

10ª Parte — Verificação do Fim do Jogo (linha 510 a 540)

- linha 510 - Se as suas chances já tiverem sido esgotadas é feita uma limpeza total da tela, caso contrário, o programa continua.
- linha 520 - Se as suas chances já tiverem sido esgotadas, a mensagem 'FIM DE JOGO' é impressa na posição (5,10) e o seu placar é impresso na posição (10,12), caso contrário, o programa continua.
- linha 530 - Se as suas chances já tiverem sido esgotadas, o programa vai para a linha 550, caso contrário, o programa continua.
- linha 540 - Manda o programa para linha 100 e reinicia a primeira fase.

11ª Parte — Finalização (linha 550 a 640)

Veja a décima parte do programa SENHA.

12ª Parte — Auto-Start do programa (linha 650 a 670)

linha 650 - Armazena em fita o programa com o nome de PORTO.

linha 660 - Roda o programa.

linha 670 - Pára o programa.

```
FIN DE JOGO
```

```
PLACAR=12
```

Figura 3

```
DESEJA JOGAR NOVAMENTE ? (S/N)
```

MODIFICAÇÕES

Modificação 1

Você pode alterar o número de chances que tem para atracar perfeitamente nos portos. Isto pode ser facilmente feito, trocando o número que a variável T assume na linha 10. Se você colocar:

```
10 LET T=5
```

poderá atracar erroneamente 5 vezes.

Modificação 2

Você pode modificar o tamanho dos três portos da parte inferior do vídeo. Para modificar o tamanho de todos os portos é só alterar a linha 30 para:

```
30 LET B = tamanho do 1º porto
```

```
40 LET C = tamanho do 2º porto
```

```
50 LET D = tamanho do 3º porto
```

Modificação 3

Você pode fazer um método de pontuação diferente. Você pode marcar 1 ponto quando conseguir atracar no porto de baixo e 1 ponto quando chegar no porto de cima. Você pode fazer isto acrescentado a linha:

```
325 LET X=X+1
```

Modificação 4

Retirando as linhas 250 e 380, você terá um efeito diferente no vídeo. Ao invés de seu navio ser um ponto "piscante", ele será um traço contínuo que mostrará todo o seu percurso tanto na ida, como na volta.

Obs.: Como o programa original tem mais de 2K, quem não tem expansão não poderia rodá-lo. Por isso nós vamos dedicar as próximas modificações para diminuir o tamanho do programa sem alterar o seu objetivo.

Modificação 5

Como primeira opção poderíamos fundir da linha 120 até a 190 em uma só. Por exemplo:

```
10 PRINT AT 10,10;"ALEPH";AT 20,  
10;"MODERNA"
```

Modificação 6

Desta vez nós iremos fundir as linhas 220 e 230. Para fundi-las nós teremos que usar as operações lógicas. Para quem não tem idéia do que se trata numa operação lógica, digite o seguinte programa.

```
10 PRINT 3>4  
20 PRINT 3=3  
30 PRINT 2+1=1+1+1  
40 PRINT 30*1=2*20  
50 PRINT 1-2=2-1
```

Após digitar RUN você deve ver na tela os seguintes números:

```
0  
1  
1  
0  
0
```


Neste programa simplesmente são executadas operações lógicas.

Exemplos:

linha 10 - Verifica se 3 é maior que 4. Se for, imprime 1, caso contrário, imprime 0.

linha 20 - Verifica se 3 é igual a 3. Se for, imprime 1, caso contrário, imprime 0.

Agora nós substituiremos as linhas 220 e 230 pela linha:

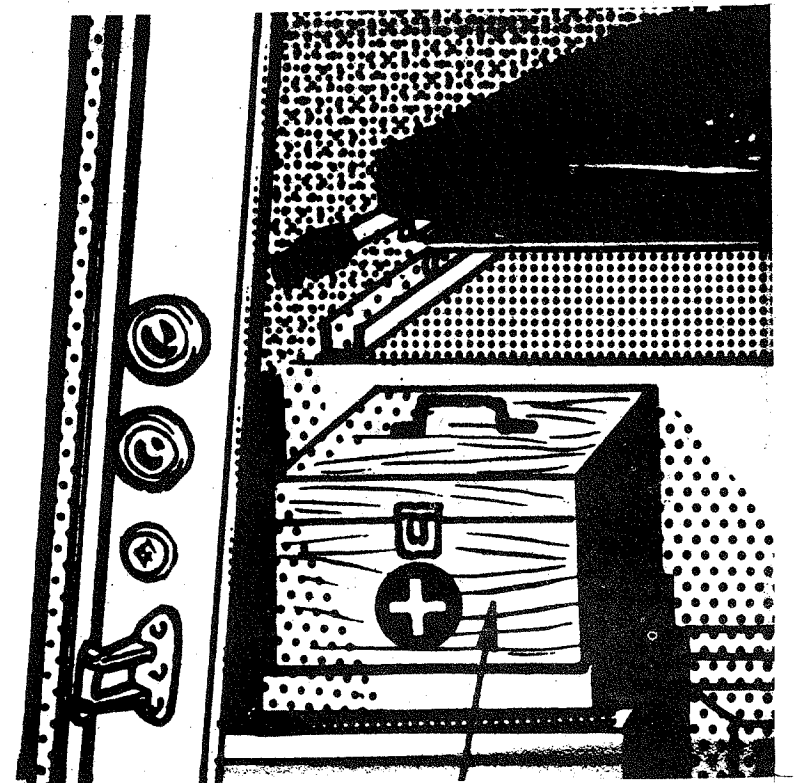
```
220 LET E=E+(INKEY#="8")-(INKEY
#="5")
```

Esta linha verifica se você está pressionando a tecla 8. Se você estiver, o computador soma 1 à variável E. Se você estiver pressionando a tecla 5 o computador subtrai 1 da variável E. Se você não pressionar nenhuma dessas teclas, a variável E continua com o valor que tinha. Esta modificação também pode ser feita nas linhas 350 e 360.

QUADRO DE VARIÁVEIS PRINCIPAIS DO PROGRAMA

- T - número de chances
- X - contadora dos pontos
- B - tamanho do primeiro porto
- C - tamanho do segundo porto
- D - tamanho do terceiro porto
- F\$ - representa o local exato onde você deve atracar
- B\$ - primeiro porto
- C\$ - segundo porto
- D\$ - terceiro porto
- E - posição inicial do barco
- Z - variável contadora da movimentação do barco

3
ambulância



3 ambulância

INSTRUÇÕES

Neste jogo você está dirigindo uma ambulância e deve recolher os feridos que estão espalhados pelas ruas deixando um sinal de perigo no local.

Toda vez que você recolher todos os feridos, mudará de bairro, e encontrará mais feridos. Cuidado, conforme você mudar de bairro o número de sinais de perigo aumenta.

Como você só atende um bairro por vez, se você tentar passar de um para outro você perderá uma ambulância. Você só tem 5 ambulâncias para salvar o maior número possível de feridos.

Figura 1



Para movimentar a ambulância use as teclas:

- 5 - para esquerda;
- 6 - para baixo;
- 7 - para cima;
- 8 - para direita.

Representações:

limite do bairro = CHR\$(128)

sinal de perigo = CHR\$(8)

ferido = 1

ambulância = 0

```
10 CLS
20 SLOW
30 LET H=0
40 LET L=5
50 LET S=0
60 LET B$=""
70 PRINT AT 20,2;"PONTOS=0";TAB
B 15;"RECORDE=";H
80 PRINT AT 21,0;"██████████"
90 FOR Z=0 TO 19
100 PRINT AT 21,0;"█";TAB 31;"█"
110 SCROLL
120 NEXT Z
130 PRINT AT 21,0;"██████████"
140 FOR Z=1 TO 10
150 PRINT AT INT (RND*19)+2,INT
(RND*30)+1;"███"
160 NEXT Z
170 FOR Z=1 TO 10
180 PRINT AT INT (RND*19)+2,INT
(RND*30)+1;"I"
190 NEXT Z
200 LET N=0
210 LET A=10
220 LET B=15
230 PRINT AT A,B;
240 LET M=PEEK (PEEK 16398+256*
PEEK 16399)
250 IF M<>0 THEN GOTO 390
260 PRINT AT A,B;"0";AT A,B;" "
270 LET A$=INKEY$
280 IF A$<>"" THEN LET B$=A$
290 IF A$="" THEN LET A$=B$
```

```
300 LET A=A+(A$="" AND A<21)-(
A$="" AND A>0)
310 LET B=B+(A$="" AND B<31)-(
A$="" AND B>0)
320 GOTO 230
330 LET L=L-1
340 PRINT AT A,B;"0"
350 IF L=0 THEN GOTO 490
360 PRINT AT 21,8;"AMBULANCIA"
";CHR$(L+156)
370 IF M=3 THEN PRINT AT A,B;"
380 IF M>=128 THEN PRINT AT A,B;
"█"
390 IF A$="" THEN LET B$=""
400 IF A$="" THEN LET B$=""
410 IF A$="" THEN LET B$=""
420 IF A$="" THEN LET B$=""
430 GOTO 270
440 PRINT AT 9,7;"BAIRROS SOCOR
RIDOS=";S/1000
450 FOR Z=0 TO 75
460 NEXT Z
470 PRINT AT A,B;" ";AT 9,7;"
480 GOTO 140
490 LET A$="FIM DE JOGO"
500 FOR C=1 TO 11
510 FOR D=1 TO 4
520 NEXT D
530 PRINT AT 10,C+8;A$(C)
540 NEXT C
550 FOR Z=0 TO 21
560 IF S>=H THEN PRINT AT Z,0;"
570 IF S<H THEN SCROLL
580 NEXT Z
590 IF S>H THEN LET H=S
600 IF S<H THEN GOTO 730
610 PRINT AT 7,11;"PARABENS";TAB
B 5;"VOCE BATEU O RECORDE";TAB 1
1;"DE HOJE";TAB 6;"ENTRE COM O S
EU NOME"
620 LET C$=""
630 LET M$=""
640 INPUT T$
650 LET C$=C$+STR$ H
660 FOR Y=1 TO LEN C$
670 LET M$=M$+CHR$(CODE C$(Y)+
128)
680 NEXT Y
```

```

690 FOR Z=0 TO 21
700 SCROLL
710 NEXT Z
720 GOTO 770
730 PRINT AT 5,8;"SEUS PONTOS="
740 PRINT AT 9,8;"RECORDE=";C#
750 FOR Z=0 TO 5
760 NEXT Z
770 PRINT AT 9,8;"RECORDE=";M#
780 FOR Z=0 TO 5
790 NEXT Z
800 PRINT "FEITO POR ";T#;AT 21
0;"DESEJA JOGAR NOVAMENTE ? (S/N)"
810 IF INKEY#="S" THEN GOTO 840
820 IF INKEY#="N" THEN GOTO 860
830 GOTO 740
840 CLS
850 GOTO 40
860 CLS
870 PRINT AT 10,10;"SEU NOME"
880 STOP
890 IF M=46 THEN GOSUB 940
900 IF M>=128 OR M=8 THEN GOTO
330
910 PRINT AT A,B;"███"
920 IF N=10 THEN GOTO 440
930 GOTO 260
940 LET N=N+1
950 PRINT AT A,B;"█"
960 LET S=S+100
970 PRINT AT 0,9;S
980 RETURN
990 SAVE "AMBULANCIA"
1000 RUN
1010 STOP

```

EXPLICAÇÃO

1ª Parte - Iniciação (linha 10 a 60)

- linha 10 - Limpa todo o conteúdo da tela.
- linha 20 - Passa o computador para o modo SLOW.
- linha 30 - Igual a variável H a zero. A variável H guardará o valor do recorde.
- linha 40 - Igual a variável S a zero. A variável S guardará o valor dos seus pontos.

- linha 50 - Igual a variável L a cinco. A variável L guardará o número de ambulâncias que você tem.
- linha 60 - A variável B\$ é igualada à direção inicial da ambulância, no caso, para a direita.

2ª Parte - Impressão da Tela (linha 70 a 190)

- linha 70 - Imprime na posição (2,20) a palavra PONTOS na mesma linha, mas na posição (15,20) é impressa a palavra RECORDE e o seu valor logo ao lado, a variável H.
- linha 80 - Imprime na posição (0,21) uma linha com 32 espaços inversos.
- linha 90 - Inicia um LOOP de 20 voltas.
- linha 100 - Imprime nas posições (0, 21) e (31, 21) espaços inversos.
- linha 110 - Realiza um SCROLL. O SCROLL consiste na subida de cada uma das linhas da tela. Por exemplo, o conteúdo da linha 21 passa para linha 20, o da linha 20 para linha 19, etc.
- linha 120 - Finaliza o LOOP.
- linha 130 - Imprime na posição (0,21) uma linha com 32 espaços inversos.
- linha 140 - Inicia um LOOP de 10 voltas.
- linha 150 - Imprime um sinal de perigo em uma posição aleatória de tela que pode variar da seguinte maneira (2 até 20, 1 até 30).
- linha 160 - Finaliza o LOOP.
- linha 170 - Inicia outro LOOP de 10 voltas.
- linha 180 - Imprime um ferido em uma posição aleatória da tela que pode variar da seguinte maneira (2 até 20, 1 até 30).
- linha 190 - Finaliza o LOOP.

3ª Parte - Verificação da Posição (linha 200 a 250)

- linha 200 - Igual a variável N a zero. A variável N guardará o número de feridos já recolhidos.
- linha 210 - Igual a variável A a dez. A variável A guardará a linha na tela que sua ambulância ocupa.
- linha 220 - Igual a variável B a quinze. A variável B guardará a coluna na tela que sua ambulância ocupa.
- linha 230 - Desloca uma espécie de cursor invisível para a posição (B,A).
- linha 240 - Utiliza a variável do sistema POSPR para saber qual o caractere que ocupa a posição citada na linha 230 e coloca o código deste caractere na variável M.
- linha 250 - Verifica se a variável M é diferente de zero, se for o programa salta para linha 890, caso contrário o programa continua.

4ª Parte — Direção da ambulância (linha 260 a 320)

- linha 260 - Imprime na posição (B,A) a letra O, que representa a ambulância, logo depois imprime em cima da letra O um espaço em branco. Isto é feito para que exista um efeito "piscante" na tela.
- linha 270 - Igual a variável A\$ a tecla pressionada. Por exemplo se você pressionou a tecla '5' a variável A\$ será igual a '5'. Se você não pressionar nenhuma tecla a variável A\$ será vazia.
- linha 280 - Verifica se alguma tecla foi pressionada, se foi esta tecla que já está armazenada em A\$ é passada para B\$, caso contrário, o programa continua.
- linha 290 - Caso nenhuma tecla tenha sido pressionada o valor da variável B\$ é passado para a variável A\$. Isto acontece, porque mesmo que você não pressione nenhuma tecla, a ambulância se movimentará na direção da última tecla pressionada. Por exemplo, se você pressionar a tecla 6 a ambulância começará a descer até que você pressione outra tecla.
- linha 300 - Dependendo da tecla pressionada, esta linha altera o valor da linha que sua ambulância ocupa, isto é feito através de operações lógicas. Veja Modificação 6, no capítulo 2.
- linha 310 - Dependendo da tecla pressionada esta linha altera o valor da coluna que sua ambulância ocupa, isto é feito através de operações lógicas.
- linha 320 - Manda o programa para a linha 230.

5ª Parte — Batida da Ambulância (linha 330 a 430)

- linha 330 - Diminui de um o valor da variável L.
- linha 340 - Imprime na posição (B,A) a letra O em vídeo inverso.
- linha 350 - Se suas ambulâncias tiverem acabado o programa vai para linha 490, caso contrário o programa continua.
- linha 360 - Imprime na posição (8, 21) a palavra 'AMBULÂNCIA=', em vídeo inverso e logo ao lado imprime o número de ambulâncias que você ainda tem.
- linha 370 - Se você bateu em um sinal de perigo é impresso na posição (B,A) um espaço em branco, caso contrário, o programa continua.
- linha 380 - Se você bateu nos limites do bairro, este limite é redesenhado, imprimindo-se na posição (B,A) um espaço inverso.
- linha 390 - Estas linhas irão inverter o sentido da sua ambulância após a batida. Por exemplo, se você vinha para a esquerda e bateu, agora sua ambulância irá andar para a direita.

linha 430 - Manda o programa para linha 270.

6ª Parte — Número de bairros socorridos (linha 440 a 480)

- linha 440 - Imprime na posição (7,9) a mensagem 'BAIRROS SOCORRIDOS' e logo ao lado o número de bairros socorridos, que é dado pela expressão S/1000.
- linha 450 - Inicia um LOOP de 76 voltas. Este LOOP é usado para dar uma pausa.
- linha 460 - Finaliza o LOOP.
- linha 470 - Imprime na posição (B,A) um espaço em branco, e imprime na posição (7,9) vinte e um espaços em branco.
- linha 480 - Manda o programa para a linha 140.

7ª Parte — Impressão da palavra FIM DE JOGO (linha 490 a 540)

- linha 490 - Igual a variável A\$ à palavra FIM DE JOGO.
- linha 500 - Inicia um LOOP de 11 voltas, no qual a variável contadora é a variável C.
- linha 510 - Inicia outro LOOP, mas este LOOP é usado apenas para dar uma pausa.
- linha 520 - Finaliza o LOOP da pausa.
- linha 530 - Imprime começando na posição (9,10) a palavra FIM DE JOGO, letra por letra.
- linha 540 - Finaliza o LOOP principal.

8ª Parte — Quebra do RECORDE (linha 550 a 640)

- linha 550 - Inicia um LOOP de 21 voltas.
- linha 560 - Verifica se seus pontos são maiores que o recorde. Se for o computador imprime na posição (O,Z) 32 espaços inversos, caso contrário, o programa continua.
- linha 570 - Caso os seus pontos sejam menor que o recorde é realizado um SCROLL, caso contrário, o programa continua.
- linha 580 - Finaliza o LOOP.
- linha 590 - Caso você tenha batido o recorde a variável H irá ter o mesmo valor da variável S.
- linha 600 - Se você não bateu o recorde o programa vai para linha 730.
- linha 610 - Imprime a mensagem "PARABÉNS VOCÊ BATEU O RECORDE DE HOJE ENTRE COM SEU NOME" no centro da tela.
- linha 620 - Igual a variável C\$ a uma string vazia. A variável C\$ irá guardar o recorde em vídeo normal.

linha 630 - Igual a variável M\$ a uma string vazia. A variável M\$ irá guardar o recorde em vídeo inverso.

linha 640 - Espera você digitar o seu nome e NEW LINE. O seu nome será guardado na variável T\$.

9ª Parte — Inversão do Recorde (linha 650 a 720)

linha 650 - Igual a variável C\$ ao recorde H. A função STR\$ faz o seguinte: ela transforma o conteúdo da variável H que é numérico em uma variável alfanumérica, no caso C\$.

linha 660 - Inicia um LOOP que terá tantas voltas quanto for o tamanho da variável C\$. A função LEN dá o tamanho de uma certa variável alfanumérica no caso C\$.

linha 670 - Armazena na variável alfanumérica M\$ o valor do recorde, mas em vídeo inverso. Isto é feito somando-se 128 ao código do recorde em vídeo normal.

linha 680 - Finaliza o LOOP.

linha 690 - Inicia um LOOP de 22 voltas.

linha 700 - Realiza um SCROLL.

linha 710 - Finaliza o LOOP.

linha 720 - Manda o programa para a linha 770.

10ª Parte — Impressão dos pontos e recorde (linha 730 a 790)

linha 730 - Imprime na posição (8, 5) a mensagem SEUS PONTOS =. Logo, ao lado é impressa a variável S que armazena seus pontos.

linha 740 - Imprime na posição (8, 9) a palavra RECORDE =. E bem ao lado o valor do recorde em vídeo normal (variável C\$).

linha 750 - Inicia um LOOP de 6 voltas para dar uma pausa.

linha 760 - Finaliza o LOOP.

linha 770 - Imprime na posição (8, 9) a palavra RECORDE = em vídeo inverso, logo ao lado é impresso o valor do recorde em vídeo inverso (variável M% verso (variável M\$)).

linha 780 - Inicia um LOOP de 6 voltas para dar uma pausa.

linha 790 - Finaliza o LOOP.

11ª Parte — Finalização (linha 800 a 880)

linha 800 - Imprime a mensagem FEITO POR, e o nome de quem fez o recorde, este nome está armazenado na variável T\$. Imprime na

posição (0, 21) a mensagem DESEJA JOGAR NOVAMENTE? (S/N).

linha 810 - Se você pressionar a letra 'S' o programa irá para linha 840.

linha 820 - Se você pressionar a letra 'N' o programa irá para linha 860.

linha 830 - Manda o programa para linha 810.

linha 840 - Limpa todo o conteúdo da tela.

linha 850 - Manda o programa para linha 40.

linha 860 - Limpa o conteúdo da tela.

linha 870 - Imprime na posição (10, 10) a palavra FIM em vídeo inverso.

linha 880 - Pára o programa.

12ª Parte — Checa onde você está (linha 890 a 980)

linha 890 - A variável M contém o código do caractere que está na posição (B, A). Se M for igual a 46 o programa vai para a linha 940.

linha 900 - Se a variável M for maior ou igual a 128 ou for igual a 8 o programa vai para linha 330.

linha 910 - Imprime na posição (B, A) o caractere 8 que representa um sinal de perigo.

linha 920 - N é a variável que armazena o número de feridos já recolhidos. Se N for igual a 10 o programa vai para a linha 440.

linha 930 - Manda o programa para linha 260.

linha 940 - Soma um a variável N.

linha 950 - Imprime na posição (B, A) a letra I em vídeo inverso.

linha 960 - Soma 100 a variável S que armazena os seus pontos.

linha 970 - Imprime na posição (9, 0) a variável S.

linha 980 - Retorna ao programa principal.

13ª Parte — Auto-Start (linha 990 a 1010)

linha 990 - Armazena o programa em fita com o nome de AMBULÂNCIA.

linha 1000 - Roda o programa.

linha 1010 - Pára o programa.

MODIFICAÇÕES

Modificação 1

Você pode alterar o número de ambulâncias no estabelecimento. Dependendo deste número você poderá fazer mais ou menos pontos. Para fazer

isto basta trocar o número que a variável L assume na linha 50. Se você colocar:

```
50 LET L=3
```

você poderá bater a ambulância três vezes.

Modificação 2

Você pode modificar o número de sinais de perigo que serão colocados no início do jogo. Isto é feito mudando o número de vezes que a linha 140 é executada. Para ter mais sinais de perigo basta alterar a linha 140 para:

```
140 FOR I=1 TO 20
```

Modificação 3

Você também pode alterar a direção inicial que a ambulância segue. Essa direção é definida pela linha 60 e basta você trocar o número que está entre aspas para 5, 6 ou 7.

QUADRO DAS VARIÁVEIS PRINCIPAIS DO PROGRAMA

- H - valor do recorde
- S - valor dos seus pontos
- L - número de ambulâncias que você tem
- B\$ - direção inicial da ambulância
- N - número de feridos recolhidos
- A - sua linha na tela
- B - sua coluna na tela
- M - código do caractere que ocupa a posição (B, A)
- A\$ - direção da ambulância durante o jogo
- C\$ - recorde em vídeo normal
- M\$ - recorde em vídeo inverso
- Z, X, C, D, Y, J - variáveis contadoras de LOOPS.

4
escavações



4 escavações

INSTRUÇÕES

Você está na Amazônia após ter encontrado um mapa de um tesouro do século X exatamente no local marcado por um X no mapa.

Durante muitos meses você fez escavações numa mina até encontrar o incrível tesouro: uma esmeralda pesando cerca de meio quilograma. Quando você já está voltando em direção à entrada, ocorre um terremoto que o soterra.

Este não é o seu maior problema, pois você tem material para fazer uma escavação até a saída. O maior problema é que o terremoto libertou um monstro pré-histórico que guardava a esmeralda.

Seu material só consegue escavar um tipo de pedra, e na mina existem dois tipos de pedra.

Tome cuidado porque essas pedras são iguais externamente!

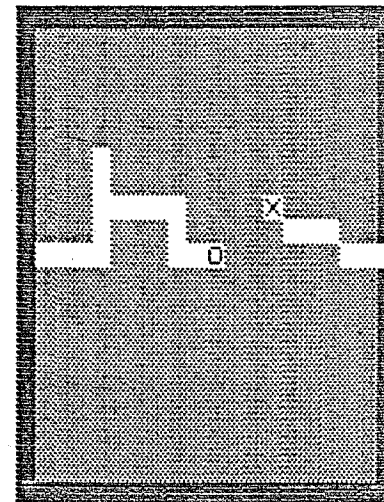


Figura 1

Para movimentar-se use os seguintes comandos:

- 5 - escava para a esquerda;
- 6 - escava para trás;
- 7 - escava para frente;
- 8 - escava para a direita.

Você será representado pela letra O, o monstro pela letra X e as pedras serão representadas pelo caractere número 8.

```

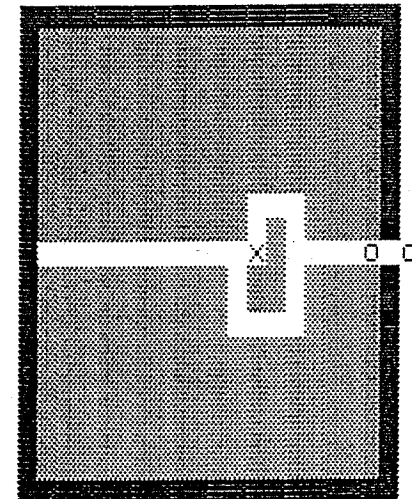
10 FAST
20 PRINT "          ESCAVACOES
5          ", "ENTRE COM O NIVE
L DE DIFICULDADE (DE 1 A 5)"
30 INPUT L
40 IF L<1 OR L>5 THEN GOTO 30
50 CLS
60 DIM A(19,19)
70 LET R=10
80 LET U=19
90 LET C=0
100 LET A$=""
110 PRINT AT 20,0;A$;AT 0,0;A$
120 FOR N=1 TO 19
130 PRINT "
140 NEXT N
150 FOR N=1 TO 100
160 LET Y=INT (RND*19)+1
170 LET X=INT (RND*19)+1
180 IF (Y=9 OR Y=10) AND (X=1 O
R X=19) THEN GOTO 160
190 LET A(Y,X)=1
200 NEXT N
210 LET Y=10
220 LET X=1
230 SLOW
240 PRINT AT 10,20;" "
250 FOR N=1 TO 6-L
260 LET C=C+1
270 LET Z=X
280 LET T=Y
290 LET Y=Y+(INKEY$="6" AND Y<1
9)-(INKEY$="7" AND Y>1)
300 LET X=X+(INKEY$="8" AND X<1
9)-(INKEY$="5" AND X>1)
310 IF Y=10 AND X=19 THEN GOTO
450
320 IF A(Y,X)=1 THEN LET X=Z
330 IF A(Y,X)=1 THEN LET Y=T
340 PRINT AT T,Z;" ";AT Y,X;"O"
350 NEXT N

```

```

360 LET E=R
370 LET S=U
380 LET V=U+(U<X)-(U>X)
390 LET R=R+(R<Y)-(R>Y)
400 IF A(R,U)=1 THEN LET R=E
410 IF A(R,U)=1 THEN LET U=S
420 PRINT AT E,S;" ";AT R,U;"X"
430 IF R=Y AND U=X THEN GOTO 47
0
440 GOTO 250
450 PRINT AT 10,19;" O";AT 1,24
;"VOCE";AT 3,23;"ESCAPOU";AT 5,2
5;"EM";AT 7,25;O;AT 9,22;"MOVIME
NTOS"
460 GOTO 480
470 PRINT AT 3,25;"VOCE";AT 5,2
5;"FOI";AT 7,23;"CAPTURADO"
480 PRINT AT 21,0;"DESEJA JOGAR
NOVAMENTE ? (S/N)"
490 IF INKEY$="S" THEN GOTO 550
500 IF INKEY$="N" THEN GOTO 520
510 GOTO 490
520 CLS
530 PRINT AT 10,10;"FIM"
540 STOP
550 CLS
560 RUN
570 SAVE "ESCAVACOES"
580 RUN

```



VOCE
ESCAPOU
EM
56
MOVIMENTOS

Figura 2

DESEJA JOGAR NOVAMENTE ? (S/N)

EXPLICAÇÃO

1ª Parte — Inicialização (linha 10 a 100)

- linha 10 - Passa o computador para o modo FAST.
- linha 20 - Imprime no início do vídeo a palavra ESCAVAÇÕES e duas linhas abaixo ENTRE COM O NÍVEL DE DIFICULDADE (DE 1 A 5).
- linha 30 - Espera que você digite o nível de dificuldade e NEW LINE e coloca este valor na variável L.
- linha 40 - Verifica se a variável L é menor que 1 ou maior que 5. Caso isto aconteça o programa volta para a linha 30, caso contrário, o programa continua.
- linha 50 - Limpa todo o conteúdo da tela.
- linha 60 - Dimensiona uma matriz quadrada A(19 x 19).
- linha 70 - Iguala a variável R a 10. A variável R guardará a linha que o monstro ocupa na tela.
- linha 80 - Iguala a variável V a 19. A variável V guardará a coluna que o monstro ocupa na tela.
- linha 90 - Iguala a variável C a zero. A variável C guardará o número de movimento já executado.
- linha 100 - Iguala a variável A\$ a 20 espaços inversos. A variável A\$ será usada para imprimir as margens inferior e superior da mina.

2ª Parte — Impressão da tela (linha 110 a 140)

- linha 110 - Imprime nas posições (0, 20) e (0, 0) a variável A\$ (margens).
- linha 120 - Inicia um LOOP de 19 voltas.
- linha 130 - Imprime o centro do labirinto que é composto de 1 espaço inverso seguido de 18 caracteres número 8 e mais 1 espaço inverso.
- linha 140 - Finaliza o LOOP.

3ª Parte — Colocação dos obstáculos e posição inicial (linha 150 a 220)

- linha 150 - Inicia um LOOP de 100 voltas.
- linha 160 - Escolhe um valor aleatório para Y variando entre 1 e 19.
- linha 170 - Escolhe um valor aleatório para X variando entre 1 e 19.
- linha 180 - Verifica se Y é igual a 9 ou 10 e se X é igual a 1 ou 19. Isto é feito para que não seja colocado um obstáculo na sua posição inicial.
- linha 190 - Iguala a variável A(Y, X) a 1, indicando que naquele local da matriz existe um obstáculo.

- linha 200 - Finaliza o LOOP.
- linha 210 - Iguala a variável Y a 10. A variável Y guardará a coluna que você ocupa na tela.
- linha 220 - Iguala a variável X a 1. A variável X guardará a linha que você ocupa na tela.

4ª Parte — Sua Movimentação (linha 230 a 350)

- linha 230 - Passa o computador para o modo SLOW.
- linha 240 - Imprime na posição (20, 10) um espaço em branco que indica a saída.
- linha 250 - Inicia um LOOP que vai ter o número de voltas variando dependendo do nível de dificuldade. Por exemplo se o nível de dificuldade for 2, o LOOP terá 4 voltas, isto quer dizer que enquanto você dá 4 passos o monstro dá apenas um.
- linha 260 - Soma 1 a variável C indicando que você fez mais um movimento.
- linha 270 - Iguala a variável Z a variável X.
- linha 280 - Iguala a variável T a variável Y. As linhas 270 e 280 existem porque, caso você faça algum movimento ilegal, a letra O que representa você será impressa na última posição legal que agora é a posição (T, Z).
- linha 290 - Esta linha verifica se você está pressionando a tecla 6, o computador soma 1 a variável Y, se você estiver pressionando a tecla 7 o computador subtrai 1 da variável Y, se você não pressionar nenhuma destas duas teclas o valor da variável Y não se altera.
- linha 300 - Esta linha verifica se você está pressionando a tecla 8. Se você estiver, o computador soma 1 a variável X, se você estiver pressionando a tecla 5 o computador subtrai 1 da variável X. Caso você não pressione nenhuma destas duas teclas o valor de X não se altera.
- linha 310 - Se a variável X for igual a 10 e Y for igual a 10 isto indica que você conseguiu sair da mina e o programa vai para a linha 450.
- linha 320 - Se a posição (Y, X) na matriz A for igual a 1 isto significa que existe um obstáculo, por isto a variável Y é igualada a última posição legal que está armazenada na variável T.
- linha 330 - Se a posição (Y, X) na matriz A for igual a 1, isto significa que existe um obstáculo, por isto a variável X é igualada a última posição legal que está armazenada na variável Z.

linha 340 - Imprime na posição (Z, T) um espaço em branco e na posição (Y, X) a letra O.

linha 350 - Finaliza o LOOP.

5ª Parte — Movimentação do Monstro (linha 360 a 440)

linha 360 - Iguale a variável E a variável R.

linha 370 - Iguale a variável S a variável V. As linhas 360 e 370 fazem, caso o monstro faça um movimento ilegal, que seja impresso na última posição legal que agora é (S, E).

linha 380 - Compara a coluna que o monstro ocupa com a sua. Se a do monstro for maior, é subtraído 1 da variável V; se a coluna do monstro for menor é somado 1 a variável V; se for igual o valor de V não se altera.

linha 390 - Compara a linha que o monstro ocupa com a sua. Se a do monstro for maior é subtraído 1 da variável R; se a linha do monstro for menor é somado 1 a variável R; se as linhas forem iguais a variável R não se altera.

linha 400 - Se a posição (R, V) na matriz A for igual a 1 isto significa que existe um obstáculo, por isto a variável R é igualada a última posição legal que está armazenada na variável E.

linha 410 - Se a posição (R, V) for igual a 1, isto significa que existe um obstáculo, por isto a variável V é igualada a última posição legal armazenada na variável S.

linha 420 - Imprime na posição (S, E) um espaço em branco e na posição (V, R) a letra X.

linha 430 - Verifica se R é igual a Y e se V é igual a X, se estas condições forem satisfeitas significa que você foi capturado pelo monstro e o programa vai para a linha 470, caso contrário, o programa continua.

linha 440 - Manda o programa para a linha 250.

6ª Parte — Escapatória (linha 450 e 460)

linha 450 - Imprime na posição (19, 10) um espaço em branco seguido da letra O; na posição (24, 1) a palavra VOCÊ; na posição (23, 3) a palavra ESCAPOU; na posição (25, 5) a palavra EM; na posição (25, 7) a variável C que guarda o número de movimentos que você fez e na posição (22, 9) a palavra MOVIMENTOS.

linha 460 - Manda o programa para a linha 480.

7ª Parte — Capturado (linha 470)

linha 470 - Imprime na posição (25, 3) a palavra VOCÊ, na posição (25, 5) a palavra FOI e na posição (23, 7) a palavra CAPTURADO.

8ª Parte — Finalização (linha 480 a 560)

linha 480 - Imprime na posição (0, 21) a mensagem DESEJA JOGAR NOVAMENTE? (S/N).

linha 490 - Se você pressionar a tecla S o programa vai para a linha 550.

linha 500 - Se você pressionar a tecla N o programa vai para a linha 520.

linha 510 - Manda o programa para a linha 490.

linha 520 - Limpa o conteúdo da tecla.

linha 530 - Imprime na posição (10, 10) a palavra FIM.

linha 540 - Pára o programa.

linha 550 - Limpa o conteúdo da tela.

linha 560 - Roda o programa.

9ª Parte — Auto-Start (linha 570 a 580)

linha 570 - Armazena o programa em fita com o nome de ESCAVAÇÕES.

linha 580 - Roda o programa.

MODIFICAÇÕES

Modificação 1

Se você incluir a linha:

```
195 PRINT AT Y,X;CHR$(128)
```

você irá tornar as pedras que você não consegue escavar diferentes das que você consegue.

Modificação 2

Você pode alterar o número de pedras que você não pode escavar e que serão postas na mina. Para fazer isto basta alterar a linha 150, para que ela fique da seguinte maneira:

```
150 FOR N = 1 TO nº de pedras desejadas
```

Modificação 3

Você pode acrescentar um comando ao jogo. Esse comando levaria você até a posição inicial caso você aperte a tecla 0 ou o botão de tiro do Joystick. Para isto você deve acrescentar as linhas:

```
305 IF INKEY$="0" THEN LET Y=10  
306 IF INKEY$="0" THEN LET X=1
```

QUADRO DAS VARIÁVEIS PRINCIPAIS DO PROGRAMA

- L - nível de dificuldade
- A - matriz que simula a tela
- R - linha do monstro na tela
- V - coluna do monstro na tela
- C - número de movimentos executados
- A\$ - margens do labirinto
- Y - coluna que você ocupa na tela
- X - linha que você ocupa na tela
- Z - variável de transição
- T - variável de transição
- E - variável de transição
- S - variável de transição

5
tiro ao alvo



5 tiro ao alvo

INSTRUÇÕES

Você acaba de entrar em um parque de diversões, muito parecido com aqueles que aparecem na televisão. A primeira barraca que você avista, não é senão, nada mais, nada menos, que a barraca do tiro ao alvo.

Nesta barraca existem 16 patinhos nadando de um lado para o outro.

Você como não pertence a nenhuma sociedade protetora de animais, empenha uma espingarda que está em cima do balcão e começa a atirar.

O seu objetivo é claro: ganhar a bonequinha que é dada a quem acertar todos os patinhos em 16 tiros.

Comando:

0 - dispara a espingarda.

Representação:

pato = CHR\$ (134)

```
10 CLS
20 SLOW
30 LET S=0
40 LET D=0
50 LET A$=""
60 PRINT "TIRO AO ALVO"
70 PRINT AT 8,14;"ATIRE"
140 PRINT AT 21,14;" ";AT 20,
147" ";AT 19,15;" "
80 PRINT AT 10,0;A$
90 IF INKEY$="0" THEN GOSUB 13
0
100 LET A$=A$(2 TO )+A$(1)
110 IF A$=""
    THEN GOTO 240
120 GOTO 80
130 IF A$(16)<>" " THEN LET D=1
140 FOR A=17 TO 10 STEP -1
150 PRINT AT A,15;"*";AT A+1,15
" "
160 NEXT A
```

```

170 IF D=1 THEN LET A$(16)="
180 IF D=1 THEN PRINT AT 8,14
ACERTOU"
190 IF D=0 THEN PRINT AT 8,14
ERROU"
200 PRINT AT 10,0);A$
210 LET D=0
220 LET S=S+1
230 RETURN
240 CLS
250 PRINT "VOCE ACERTOU TODOS
OS PATINHOS EM ";S;" TIROS."
260 PRINT AT (21,0);"DESEJA JOGAR
NOVAMENTE ? (S/N)"
270 IF INKEY#="S" THEN RUN
280 IF INKEY#="N" THEN GOTO 300
290 GOTO 270
300 CLS
310 PRINT AT 10,10);"FIM"
320 STOP
330 SAVE "TIRO AO ALVO"
340 RUN

```

EXPLICAÇÃO

1ª Parte – Inicialização (linha 10 a 50)

- linha 10 - Limpa todo o conteúdo da tela.
- linha 20 - Passa o computador para o modo SLOW.
- linha 30 - Igual a variável S a zero. A variável S guardará o número de tiros que você deu.
- linha 40 - Igual a variável D a zero. A variável D irá dizer se você acertou ou errou o alvo.
- linha 50 - Igual a variável A\$ a 16 vezes um caractere 134 seguido de um espaço em branco. A variável A\$ será a representação da linha de patos.

2ª Parte – Impressão da tela (linha 60 a 80)

- linha 60 - Imprime no início da tela o título TIRO AO ALVO e depois imprime na posição (14, 8) a palavra ATIRE.
- linha 70 - Imprime na posição (14, 21) três espaços inversos depois imprime na posição (20, 14) um caractere 133 seguido de um espaço inverso seguido pelo caractere 5. Finalmente é impresso um espaço inverso na posição (15, 19). Esta linha desenha sua espingarda.
- linha 80 - Imprime na posição (0, 10) a variável A\$.

3ª Parte – Rotação dos Alvos (linha 90 a 120)

- linha 90 - Verifica se você pressionou a tecla 0, se esta condição for verdadeira, o programa chama a rotina de tiro na linha 130, caso contrário, o programa continua.
- linha 100 - Esta linha é que realmente dá o efeito de rotação dos alvos. Ela funciona da seguinte maneira: a instrução (2 TO) ignora o primeiro caractere da variável A\$ mas este caractere é somado no final da mesma variável dando o efeito de rotação.
- linha 110 - Se a variável A\$ for constituída apenas de 32 espaços em branco, significa que você já acertou todos os patos e o programa vai para a linha 240.
- linha 120 - Manda o programa para a linha 80.

4ª Parte – Rotina de Tiro (linha 130 a 230)

- linha 130 - Verifica se a posição 16 da variável A\$ é diferente de um espaço em branco, o que significa que você atirou e acertou um pato, e neste caso a variável A é igualada a 1.
- linha 140 - Inicia um LOOP que começa no 17 e acaba no 10. Para fazer um LOOP regressivo basta acrescentar a função STEP -1. Esta função altera o passo do LOOP.
- linha 150 - Imprime na posição (15, A) um asterisco e logo a seguir imprime um espaço em branco na posição (15, A + 1). Isto é feito para que haja um de tiro.
- linha 160 - Finaliza o LOOP regressivo.
- linha 170 - Se a variável D for igual a 1, a posição da variável A\$ (16) é igualada a um espaço em branco.
- linha 180 - Se a variável D for igual a 1 é impressa a palavra ACERTOU na posição (14, 8).
- linha 190 - Se a variável D for igual a zero é impresso na posição (14, 8) a palavra ERROU, seguida de dois espaços em branco.
- linha 200 - Imprime na posição (0, 10) a variável A\$.
- linha 210 - Zera a variável D.
- linha 220 - Soma 1 a variável S.
- linha 230 - Retorna ao programa principal.

5ª Parte – Finalização (linha 240 a 320)

- linha 240 - Limpa o conteúdo da tela.
- linha 250 - Imprime no começo da tela a mensagem "VOCÊ ACERTOU

TODOS OS PATINHOS EM", imprime também o valor da variável S e logo depois a palavra TIROS.

linha 260 - Imprime na posição (0, 21) a mensagem "DESEJA JOGAR NOVAMENTE" ? (S/N).

linha 270 - Se você pressionar a tecla S o programa roda novamente.

linha 280 - Se você pressionar a tecla N o programa vai para a linha 300.

linha 290 - Manda o programa para a linha 270.

linha 300 - Limpa o conteúdo da tela.

linha 310 - Imprime na posição (10, 10) a palavra FIM em vídeo inverso.

linha 320 - Pára o programa.

6ª Parte — Auto-Start (linha 330 a 340)

linha 330 - Armazena o programa em fita com o nome de "TIRO AO ALVO".

linha 340 - Roda o programa.

MODIFICAÇÕES

Modificação 1

Você pode alterar a tecla que dispara o tiro, alterando a linha 90. Se você alterar a linha 90 para:

```
90 IF INKEY$="F" THEN GOSUB 13
```

a espingarda atirárá quando você pressionar a tecla F.

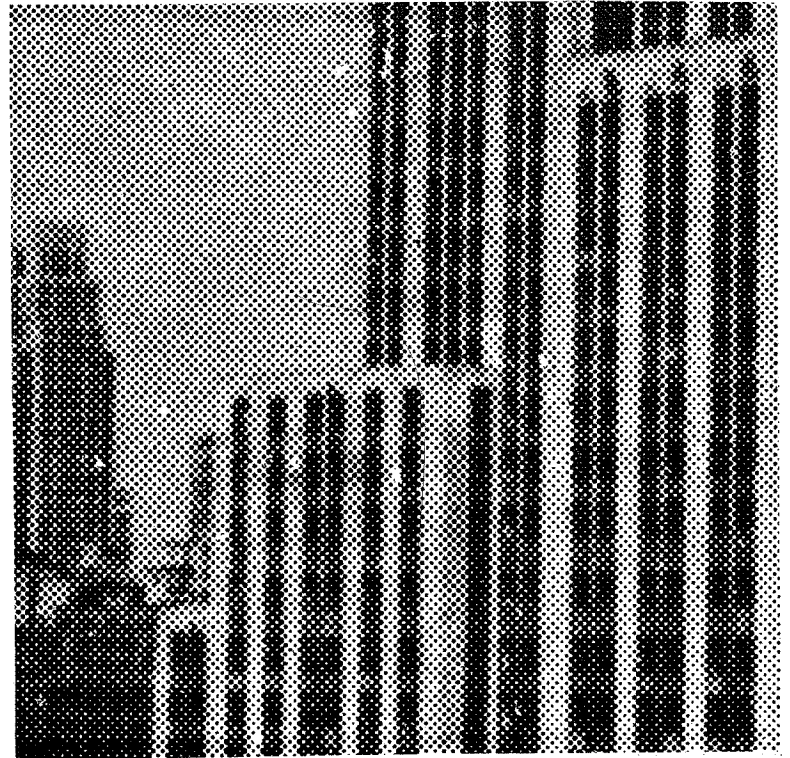
Modificação 2

Você pode modificar o caractere que simula o tiro apenas trocando o asterisco da linha 150 pelo caractere desejado. Tente trocá-lo pelo ponto final.

QUADRO DAS VARIÁVEIS PRINCIPAIS DO PROGRAMA

- S - variável contadora de tiros
- D - indicador de acerto do alvo
- A\$ - representação dos patos
- A - indicadora da posição do tiro.

6
prédios



6 prédios

INSTRUÇÕES

Neste jogo você é um simples operário de uma construção, mas será uma peça extremamente importante na reconstrução de vários prédios, isto porque o engenheiro que projetou-os cometeu um erro que obriga-o a refazer as estruturas para evitar um terrível acidente.

A sua missão é a seguinte: você deve arrancar todos os pregos que seguram a estrutura do prédio para que ele seja reconstruído.

O principal perigo é que os prédios estão sendo construídos no "Morro dos Ventos Uivantes" e você precisa tomar cuidado para não ser carregado pelo vento.

Para ajudá-lo nesta difícil missão você conta com a ajuda de um aparelho que mede a intensidade do vento. Enquanto este aparelho registrar uma intensidade entre 0 a 7 fique tranquilo, mas quando ele começar a indicar 8 ou 9 tome muito cuidado e ande sempre na direção oposta a do vento.

Apenas um lembrete: como todos os prédios, este é constituído de vários andares e para você mudar de um andar pode usar as escadas ou o elevador no lado direito do prédio.

A direção do vento é sempre da sua esquerda para a direita.

Comandos:

5 - para a esquerda;

6 - para baixo;

7 - para cima;

8 - para direita;

ou pelo joystick.

Representações:

Você - *

Prego martelado - .

Prego arrancado - ,

Escada - :

Elevador - CHR\$ 136 + CHR\$ 136


```

10 LET LL=1
20 CLS
30 LET DE=0
40 LET A=20
50 LET B=INT (RND*80)
50 LET L$=" "
70 LET L1=5
80 LET L2=30
90 LET U=26
100 LET N=1
110 IF LL=1 THEN LET S=0
120 LET W=INT (RND*10)
130 POKE 16418,1
140 PRINT AT 22,14;W
150 GOSUB 590
160 IF RND>.4 THEN LET W=W+(INT
(RND*3)-1)
170 LET W=W+(W<0)-(W>9)
180 PRINT AT 22,14;W
190 PRINT AT A,B;CHR$ U;AT L1,L
200 IF W=8 THEN LET B=B+1
210 IF W=9 THEN LET B=B+2
220 LET L1=L1+N
230 PRINT AT L1,L2;L$
240 IF U=136 THEN LET A=L1
250 IF L1=20 THEN LET N=-1
260 IF L1=4 THEN LET N=1
270 LET A=A+(INKEY$="8")-(INKEY
$="7")
280 LET B=B+(INKEY$="8")-(INKEY
$="5")
290 PRINT AT A,B;
300 LET U=PEEK (PEEK 16398+256*
PEEK 16399)
310 IF U=27 THEN LET S=S+W
320 IF U=27 THEN LET DE=DE+1
330 IF U=27 THEN LET U=26
340 IF U=128 OR U=0 THEN GOTO 3
50
350 PRINT "*"
360 IF DE=82 THEN GOTO 540
370 GOTO 160
380 PRINT AT 21,0;" "
390 LET T=1
400 FOR I=A TO 20
410 PRINT AT I,B;"*";AT I-1,B;"
";AT 21,T;" "
420 LET T=T+1
430 NEXT I
440 PRINT AT 20,10;"<<<<*>>>>";AT
21,10;"SALARIO"
450 PRINT AT 0,0;"O SALARIO",

```

```

E DE "CR$";S",00"
460 POKE 16418,2
470 PRINT AT 21,0;"DESEJA JOGAR
NOVAMENTE ? (S/N) "
480 IF INKEY$="S" THEN RUN
490 IF INKEY$="N" THEN GOTO 510
500 GOTO 480
510 CLS
520 PRINT AT 10,10;"FIM"
530 STOP
540 LET LL=LL+1
550 PRINT "PREDIO=";LL
560 FOR I=1 TO 20
570 NEXT I
580 GOTO 20
590 LET A$="....."
600 LET B$="....."
610 PRINT AT 21,0;B$;AT 20,0;A$
;AT 21,30;" ";AT 13,2;B$( TO 28
);AT 12,2;A$( TO 28);AT 18,11;B$
( TO 8);AT 17,11;A$( TO 8);AT 8,
7;B$( TO 23);AT 7,7;A$( TO 23);A
T 5,19;B$( TO 11);AT 4,19;A$( TO
11)
620 FOR Z=12 TO 20
630 PRINT AT Z,2;" ";AT Z,10;"
";AT Z,19;" ";AT Z,26;" "
640 NEXT Z
650 FOR Z=7 TO 12
660 PRINT AT Z,6;" ";AT Z,22;"
";AT Z,29;" "
670 NEXT Z
680 FOR Z=4 TO 7
690 PRINT AT Z,19;" ";AT Z,25;"
"
700 NEXT Z
710 SLOW
720 RETURN
730 SAVE "PREDIOS"
740 RUN

```

EXPLICAÇÕES

1ª Parte – Inicialização (linha 10 a 120)

- linha 10 - Igual a variável LL a 1. A variável LL guardará de quantos prédios você já arrancou todos os pregos.
- linha 20 - Limpa todo o conteúdo da tela.

- linha 30 - Iguale a variável DE a zero. A variável DE guardará quantos pregos já foram arrancados.
- linha 40 - Iguale a variável A a 20. A variável A guardará a linha que você ocupa na tela.
- linha 50 - Escolhe um valor aleatório para variável B entre 0 e 29. A variável B guardará a coluna que você ocupa na tela.
- linha 60 - Iguale a variável L\$ a 2 caracteres 136. A variável L\$ é o desenho do elevador.
- linha 70 - Iguale a variável L1 a 5. A variável L1 guardará a linha que o elevador ocupa.
- linha 80 - Iguale a variável L2 a 30. A variável L2 guardará a coluna que o elevador ocupa.
- linha 90 - Iguale a variável U a 26. A variável U guardará o caractere que será impresso na posição (B, A). O caractere 26 no caso é o *.
- linha 100 - Iguale a variável N a 1. A variável N guardará quantos andares o elevador irá subir ou descer.
- linha 110 - Se for o primeiro prédio do qual você está arrancando pregos a variável S é igualada a zero. A variável S guardará o seu salário.
- linha 120 - Escolhe um valor aleatório para a variável W entre 0 e 9. A variável W guardará a intensidade do vento.

2ª Parte – Impressão do Prédio (linha 130 a 150)

- linha 130 - Esta linha possibilita que o vídeo do seu computador tenha uma linha a mais.
- linha 140 - Imprime na posição (14, 22) a variável W.
- linha 150 - Chama a sub-rotina de impressão do prédio na linha 590.

3ª Parte – Verificação do Aparelho de Vento (linha 160 a 210)

- linha 160 - A função RND gera um número aleatório entre 0 e 1. Se este número for maior que 0,4, então a intensidade do vento é alterada, caso contrário, ela continua a mesma. A intensidade do vento é alterada da seguinte maneira, pode-se acrescentar 1 ou subtrair a da variável W.
- linha 170 - Altera o valor da variável W usando operações lógicas lógicas. Se o valor de W for menor que 1 é somado 1, se for maior do que 9 é subtraído 1.
- linha 180 - Imprime na posição (14, 22) a variável W.

- linha 190 - Imprime na posição (B, A) o caractere que tem o código igual a da variável U e na posição (L2, L1) dois espaços em branco que limpa a antiga posição do elevador.
- linha 200 - Se a variável W for igual a 8 é somado 1 a variável B. Isto significa que você está sendo carregado pelo vento 1 posição.
- linha 210 - Se a variável W for igual 9 é somado 2 a variável B. Isto significa que você foi carregado 2 posições pelo vento.

4ª Parte – Movimento do Elevador (linha 220 a 230)

- linha 220 - Soma N a variável L1 se N for igual a 1 o elevador está descendo e se N for igual a -1 o elevador está subindo.
- linha 230 - Imprime na posição (L2, L1) a variável L\$.

5ª Parte – Verificação (linha 240 a 260)

- linha 240 - Se a variável U for igual a 136 significa que você está dentro do elevador. Se esta condição for verdadeira a variável A deve ser igual a L1, lembrando que A é a sua linha na tela e L1 é a linha do elevador na tela.
- linha 250 - Se a variável L1 for igual a 20, isto significa que o elevador chegou ao térreo e a variável N deve ser igual a -1 para ele começar a subir.
- linha 260 - Se a variável L1 for igual a 4, significa que o elevador chegou ao último andar e a variável N deve valer 1 para ele começar a descer.

6ª Parte – Sua Movimentação (linha 270 a 280)

- linha 270 - Esta linha verifica se você está pressionando a tecla 6. Se você estiver, é somado 1 a variável A. Se você estiver pressionando a tecla 7 é subtraído 1 da variável A. Se você não pressionar nenhuma das duas teclas o valor da variável A não se altera.
- linha 280 - Esta linha verifica se você está pressionando a tecla 6. Se estiver, é somado 1 a variável B. Se você estiver pressionando a tecla 5 é subtraído 1 da variável B. Se você não pressionar nenhuma destas teclas o valor de B não se altera.

7ª Parte – Verificação do seu Movimento (linha 290 a 340)

- linha 290 - Desloca uma espécie de cursor invisível para a posição (B, A).
- linha 300 - Utiliza a variável do sistema POSPR para saber qual o caractere

que ocupa a posição citada na linha 290 e coloca o código deste caractere na variável U.

linha 310 - Se a variável U for igual a 27 (ponto) são somados tantos cruzeiros quantos estiver marcando o aparelho que marca a intensidade do vento.

linha 320 - Se a variável U for igual a 27 (ponto) é somado 1 a variável DE.

linha 340 - Se a variável U for igual a 128 ou a 0 significa que você caiu do prédio e o programa vai para a linha 380.

8ª Parte — Impressão e Verificação do Fim do Prédio (linha 350 a 370).

linha 350 - Imprime um * (você).

linha 360 - Se a variável DE for igual a 82 significa que você arrancou todos os pregos deste prédio e o programa vai para a linha 540.

linha 370 - Manda o programa para a linha 160.

9ª Parte — Fim do Jogo (linha 380 a 450)

linha 380 - Imprime na posição (0, 21) a letra em vídeo inverso.

linha 390 - Igual a variável T a 1. A variável T funcionará como contador de quantas letras I na linha 410 serão impressas.

linha 400 - Inicia um LOOP que começa no número da linha que você ocupa até 20.

linha 410 - Imprime na posição (B, I) um asterisco, na posição (B, I - 1) um espaço em branco. Isto dá a impressão que você está caindo do prédio. Imprime na posição (T, 21) a letra I em vídeo inverso.

linha 420 - Soma 1 a variável T.

linha 430 - Finaliza o LOOP.

linha 440 - Imprime na posição (10, 20) o desenho * e na posição (10, 21) a palavra BUUMM em vídeo inverso.

linha 450 - Imprime na posição (0, 0) a palavra SALÁRIO; na posição (0, 2) a frase E DE e na posição (0, 4) o símbolo Cr\$ seguido pelo valor da variável S.

10ª Parte — Finalização (linha 460 a 530)

linha 460 - Tira o efeito da linha 130.

linha 470 - Imprime na posição (0, 21) a mensagem "DESEJA JOGAR NOVAMENTE" ? (S/N).

linha 480 - Se você pressionar a tecla S, o programa roda novamente.

linha 490 - Se você pressionar a tecla N, o programa vai para a linha 510.

linha 500 - Manda o programa para a linha 480.

linha 510 - Limpa todo o conteúdo da tela.

linha 520 - Imprime na posição (10, 10) a palavra "FIM" em vídeo inverso.

linha 530 - Pára o programa.

11ª Parte — Mudança de Prédio (linha 540 a 580)

linha 540 - Soma 1 a variável LL.

linha 550 - Imprime o prédio no qual você está.

linha 560 - Inicia um LOOP de 100 voltas.

linha 570 - Finaliza o LOOP. Este LOOP serve apenas para dar uma pausa no programa.

linha 580 - Manda o programa para a linha 20.

12ª Parte — Rotina de Impressão do Prédio (linha 590 a 720)

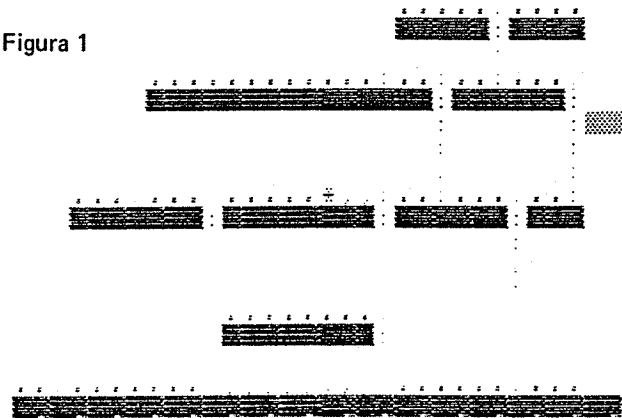
Veja Apêndice B

13ª Parte — Auto-Start do Programa (linha 730 a 740)

linha 730 - Armazena o programa em fita com o nome de PRÉDIOS.

linha 740 - Roda o programa.

Figura 1



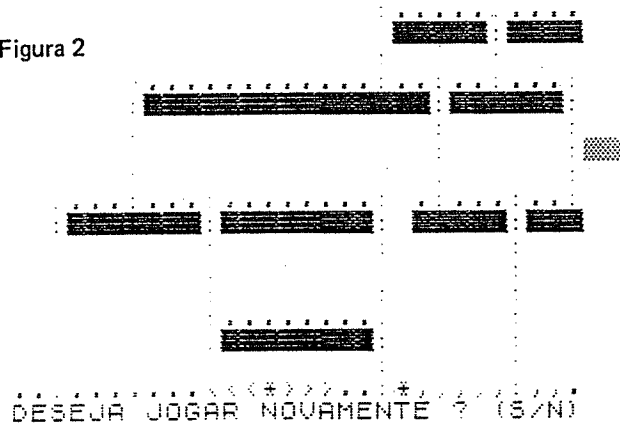
MODIFICAÇÕES

Veja Apêndice B

QUADRO DAS VARIÁVEIS PRINCIPAIS DO PROGRAMA

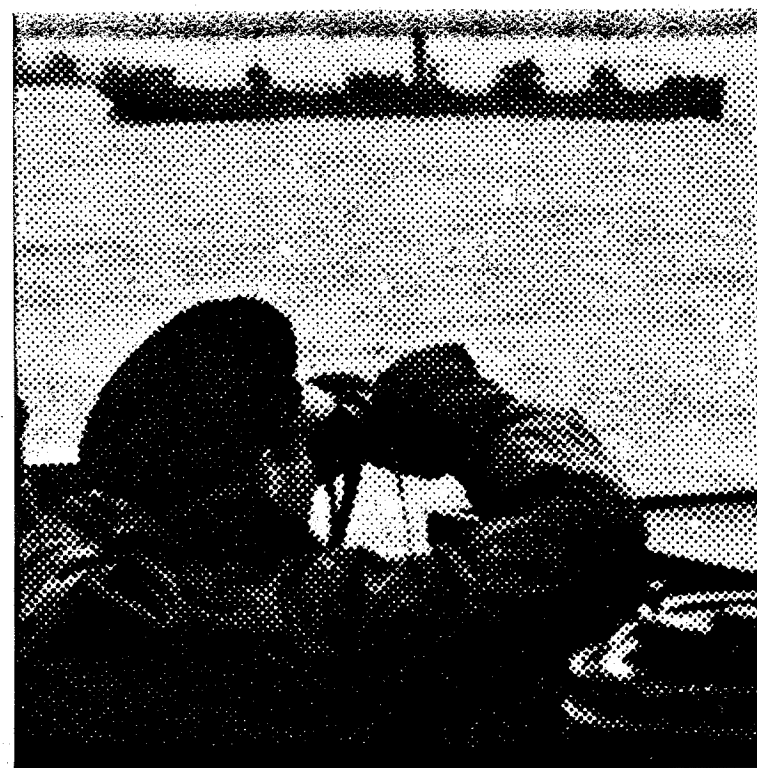
- LL - prédios com todos os pregos arrancados
- DE - pregos já arrancados
- A - sua linha na tela
- B - sua coluna na tela
- L\$ - representação do elevador
- L1 - linha do elevador na tela
- L2 - coluna do elevador na tela
- U - caractere que será impresso na posição (B, A)
- N - andares que o elevador subirá ou descerá
- S - valor do seu salário
- W - intensidade do vento
- T, I, Z - variáveis contadoras de LOOPS.

Figura 2



7

ataque submarino



7 ataque submarino

INSTRUÇÕES

Você acaba de entrar em uma batalha naval no meio do Oceano Índico, você é o comandante de um submarino com um arsenal limitado a 30 torpedos. Com estes 30 torpedos, você tentará destruir o maior número de inimigos possível.

Só que existe um problema com o seu radar: ele só mostra a posição dos inimigos durante 2 segundos e durante este tempo você deverá memorizar a posição dos mesmos.

Logo depois será mostrada uma espécie de tabela no seu computador de bordo. Você verificará em sua memória a posição dos inimigos e entrará com as coordenadas da seguinte forma:

coluna, linha

Exemplos:

A1

B7

M9

Você tem três tipos de inimigos para destruir, os hidro-aviões que valem 30 pontos, os cruzadores que valem 20, e as fragatas que valem apenas 10 pontos. O jogo acaba quando acabarem os torpedos do arsenal.

```
10 CLS
20 DIM A(17,29)
30 LET P=0
40 FAST
50 LET C=168
60 LET U=1
70 GOSUB 530
80 LET C=171
90 LET U=2
100 GOSUB 530
110 LET C=173
```

```

120 LET V=3
130 GOSUB 530
140 SLOW
150 PAUSE 100
160 CLS
170 GOSUB 420
180 LET G=30
190 PRINT AT 20,0;"PLACAR=";P
200 INPUT B$
210 IF LEN B$(>2 OR CODE B$+COD
E B$(2)>89 THEN GOTO 200
220 LET L=(VAL B$(2))*2-1
230 LET T=(CODE B$(1))*2-75
240 LET B=A(L,T)
250 IF B=1 THEN GOTO 330
260 IF B=2 THEN GOTO 360
270 IF B=3 THEN GOTO 390
280 PRINT AT L,T;"X"
290 LET G=G-1
300 PRINT AT 20,13;"SOBRAM ";G;
" TORPEDOS "
310 IF G=0 THEN GOTO 690
320 GOTO 200
330 LET C=168
340 LET V=20
350 GOTO 600
360 LET C=171
370 LET V=10
380 GOTO 600
390 LET C=173
400 LET V=30
410 GOTO 600
420 LET N=1
430 FOR L=1 TO 17 STEP 2
440 PRINT AT L,0;N
450 LET N=N+1
460 NEXT L
470 LET D=38
480 FOR T=1 TO 29 STEP 2
490 PRINT AT 0,T;CHR$ D
500 LET D=D+1
510 NEXT T
520 RETURN
530 FOR E=1 TO 10
540 LET T=INT (RND*14)*2+1
550 LET L=INT (RND*9)*2+1
560 PRINT AT L,T;CHR$ C
570 LET A(L,T)=V
580 NEXT E
590 RETURN
600 PRINT AT L,T;CHR$ C
610 PRINT AT 21,0;"AFUNDOU..."

```

```

620 LET P=P+V
630 PRINT AT 20,0;"PLACAR=";P
640 LET G=G-1
650 PRINT AT 20,13;"SOBRAM ";G;
" TORPEDOS "
660 PRINT AT 21,0;" "
670 IF G=0 THEN GOTO 690
680 GOTO 200
690 PRINT AT 21,0;"DESEJA JOGAR
NOVAMENTE ? (S/N) "
700 IF INKEY$="S" THEN RUN
710 IF INKEY$="N" THEN GOTO 740
720 GOTO 700
730 CLS
740 PRINT AT 10,10;"FIM"
750 STOP
760 SAVE "ATAQUE SUBMARIN"
770 RUN

```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		■													
2				■											
3			■	X											
4					■	■		■							
5					X	X									
6					■	X	X								
7															
8															
9															

Figura 1

PLACAR=140 SOBRAM 18 TORPEDOS

EXPLICAÇÕES

1ª Parte – Iniciação (linha 10 a 40)

linha 10 - Limpa o conteúdo da tela.

linha 20 - Dimensiona uma matriz A de 17 por 29.

linha 30 - Zera a variável P que contará os seus pontos.

linha 40 - Passa o computador para o modo FAST.

2ª Parte — Impressão dos inimigos (linha 50 a 160)

linha 50 - Igual a variável C a 168. A variável C guardará o valor do caractere a ser impresso na rotina de impressão, no caso um C em vídeo inverso.

linha 60 - Igual a variável V a 1. O número 1 será a representação de um cruzador na matriz.

linha 70 - Chama a sub-rotina de impressão na linha 530.

linha 80 - Igual a variável C a 171. A variável C guardará o valor do caractere a ser impresso na rotina de impressão, no caso um F em vídeo inverso.

linha 90 - Igual a variável V a 2. O número 2 será a representação de uma fragata na matriz.

linha 100 - Chama a sub-rotina de impressão na linha 530.

linha 110 - Igual a variável C a 173. A variável C guardará o valor do caractere a ser impresso na rotina de impressão, no caso um H em vídeo inverso.

linha 120 - Igual a variável V a 3. O número 3 representa um hidro-avião na matriz.

linha 130 - Chama a sub-rotina de impressão na linha 530.

linha 140 - Passa o computador para o modo SLOW.

linha 150 - Dá uma pausa de cerca de 2 segundos.

linha 160 - Limpa todo o conteúdo da tela.

3ª Parte — Impressão das coordenadas (linha 170 a 190)

linha 170 - Chama a sub-rotina de impressão das coordenadas na linha 420.

linha 180 - Igual a variável G a 30. A variável G guardará o número de torpedos que ainda lhe restam.

linha 190 - Imprime na posição (0, 20) a palavra PLACAR = seguida do valor da variável P.

4ª Parte — Entrada das Coordenadas (linha 200 a 280)

linha 200 - Espera que você entre com as coordenadas e colocá-as na variável B\$.

linha 210 - Verifica se a maneira que você entrou com os dados foi válida.

linha 220 - Transforma o valor da parte numérica da variável B\$ multiplicada por 2 e subtrai 1, o resultado é colocado na variável L.

linha 230 - Multiplica a parte alfanumérica da variável B\$ por 2 e subtrai 75, coloca o resultado na variável T.

linha 240 - Verifica qual o valor que ocupa a posição (L, T) na matriz A e coloca este valor na variável B.

linha 250 - Se o valor de B for igual a 1, isto significa que você acertou um cruzador e o programa vai para a linha 330.

linha 260 - Se o valor da variável B for igual a 2, isto significa que você acertou uma fragata e o programa vai para a linha 360.

linha 270 - Se o valor da variável B for igual a 3, isto significa que você acertou um hidro-avião e o programa vai para a linha 390.

linha 280 - Imprime na posição (T, L) uma letra X indicando que você errou o tiro.

5ª Parte — Verificação FIM de Jogo (linha 290 a 320)

linha 290 - Subtrai 1 da variável G, indicando que você acaba de usar o torpedo.

linha 300 - Imprime na posição (13, 20) a palavra SOBRAM seguido pelo valor da variável G e pela palavra TORPEDOS.

linha 310 - Verifica se acabaram os torpedos do seu arsenal, caso tenham acabado o programa vai para a linha 690.

linha 320 - Manda o programa para a linha 200.

6ª Parte — Tiro Certo (linha 330 a 410)

linha 330 - Igual a variável C a 168 que é o código do inimigo que você acertou, no caso um cruzador.

linha 340 - Igual a variável V a 20, que é o número de pontos que o cruzador vale.

linha 350 - Manda o programa para a linha 600.

linha 360 - Igual a variável C a 171 que é o código do inimigo que você acertou, no caso uma fragata.

linha 370 - Igual a variável V a 10, que é o número de pontos que a fragata vale.

linha 380 - Manda o programa para a linha 600.

linha 390 - Igual a variável C a 173 que é o código do inimigo que você acertou, no caso um hidro-avião.

linha 410 - Igual a variável V a 30, que é o número de pontos que o hidro-avião vale.

linha 410 - Manda o programa para a linha 600.

7ª Parte — Rotina de Impressão das Coordenadas (linha 420 a 520)

- linha 420 - Iguala a variável N a 1. A variável N guardará os valores das coordenadas numéricas que serão impressas na linha 440.
- linha 430 - Inicia um LOOP de 9 voltas que começa no 1 e vai até o 17 pulando de 2 em 2.
- linha 440 - Imprime na posição (0, L) a variável N.
- linha 450 - Soma 1 a variável N.
- linha 460 - Finaliza o LOOP.
- linha 470 - Iguala a variável D a 38. A variável D guardará o código das coordenadas alfanuméricas impressas na linha 490.
- linha 480 - Inicia um LOOP de 15 voltas que começa no 1 e vai até 29 pulando de 2 em 2.
- linha 490 - Imprime na posição (T, 0) o caractere que tem o valor na variável D.
- linha 500 - Soma 1 a variável D.
- linha 510 - Finaliza o LOOP.
- linha 520 - Retorna ao programa principal.

8ª Parte — Rotina de colocação aleatória do Inimigo (linha 530 a 590)

- linha 530 - Inicia um LOOP de 10 voltas.
- linha 540 - Iguala a variável T a um valor aleatório que pode ser qualquer número inteiro e ímpar entre 1 e 27.
- linha 550 - Iguala a variável L a um valor aleatório que pode ser qualquer número inteiro e ímpar entre 1 e 17.
- linha 560 - Imprime na posição (T, L) o caractere cujo código está armazenado na variável C. A variável C foi definida no programa principal.
- linha 570 - Faz a posição (L, T) na matriz A ter um valor igual ao da variável V. A variável V foi definida no programa principal.
- linha 580 - Finaliza o LOOP.
- linha 590 - Retorna ao programa principal.

9ª Parte — Naufrágio do Inimigo (linha 600 a 680)

- linha 600 - Imprime na posição (T, L) o caractere cujo código está armazenado na variável C. A variável C foi definida no programa principal.
- linha 610 - Imprime na posição (0, 21) a palavra AFUNDOU . . .
- linha 620 - Soma a variável P o valor da variável V.

- linha 630 - Imprime na posição (0, 20) a palavra PLACAR = seguido do valor da variável P.
- linha 640 - Subtrai 1 da variável G.
- linha 650 - Imprime na posição (13, 20) a palavra SOBRAM seguido pelo valor da variável G e pela palavra TORPEDOS.
- linha 660 - Imprime na posição (0, 21) 10 espaços em branco, para apagar a mensagem impressa na linha 610.
- linha 670 - Verifica se acabaram os torpedos do seu arsenal. Caso eles tenham acabado, o programa vai para a linha 690.
- linha 680 - Manda o programa para a linha 200.

10ª Parte — Finalização (linha 690 a 750)

- linha 690 - Imprime na posição (0, 21) a mensagem "DESEJA JOGAR NOVAMENTE" ? (S/N).
- linha 700 - Se você pressionar a tecla S o programa roda novamente.
- linha 710 - Se você pressionar a tecla N o programa vai para a linha 730.
- linha 720 - Manda o programa para a linha 700.
- linha 730 - Limpa todo o conteúdo da tela.
- linha 740 - Imprime na posição (10, 10) a palavra FIM em vídeo inverso.
- linha 750 - Pára o programa.

11ª Parte — Finalização (linha 760 a 770)

- linha 760 - Armazena o programa em fita com o nome de "ATAQUE SUB-MARINO".
- linha 770 - Roda o programa.

MODIFICAÇÕES

Modificação 1

Você pode alterar o número de torpedos que possui no início do jogo, alterando a linha 180:

180 LET G = número de torpedos desejado

Modificação 2

Você pode modificar o tempo que os inimigos permanecem no radar, alterando a linha 150. Quanto maior for o número que você colocar após o comando PAUSE, maior será o tempo que eles permanecerão no radar.

QUADRO DAS VARIÁVEIS DO PROGRAMA

- A - matriz que é uma cópia da tela
- P - seu placar
- C - valor do caractere que é impresso na rotina de impressão
- V - representações dos inimigos na matriz A
- G - número de torpedos no arsenal
- B\$ - coordenadas
- L - transforma as coordenadas entradas em valores da matriz
- T - transforma as coordenadas entradas em valores da matriz
- B - valor que ocupa a posição (L, T) na matriz A
- D - código das coordenadas alfanuméricas
- N - valor das coordenadas numéricas
- E - variável contadora de LOOP.

apêndice A

Neste apêndice nós apresentamos o programa SENHA com todas as modificações que foram sugeridas no primeiro capítulo.

```
10 PRINT AT 0,12;"SENHA"
11 PRINT "VOCE QUER NUMEROS
REPETIDOS ?"
12 INPUT A$
13 PRINT A$
14 PRINT "QUANTOS ALGARISMOS
TERA O NUMEROQUE EU VOU ESCOLHE
R ?"
15 INPUT X
16 PRINT X
17 PRINT "EU POSSO ESCOLHER
NUMEROS ENTRE 0 E ?"
18 INPUT B$
19 PRINT B$
20 LET Y=CODE B$
21 PRINT "QUANTAS CHANCES VO
CE QUER TER PARA ACERTAR O NUM
ERO ?"
22 INPUT W
23 PRINT "ESTA BEM, AGUARDE
ENQUANTO EU ESCOLHO O NUME
RO."
24 PAUSE 100
25 CLS
26 LET L$=""
27 FOR J=1 TO X
28 LET L$=L$+" "
29 NEXT J
30 DIM N$(X)
35 DIM G$(X)
40 DIM U$(X)
50 LET C=0
60 RAND
65 FAST
70 FOR I=1 TO X
80 LET N$(I)=CHR$ INT (RND*(26-
26)+26)
```

```

85 FOR J=1 TO X
86 IF A$(1)="N" THEN IF N# I =
N$(J) AND I<>J THEN GOTO 80
87 NEXT J
88 NEXT I
100 PRINT AT 4,0,
110 LET C=C+1
120 IF C=U+1 THEN GOTO 360
125 SLOW
130 INPUT G$
140 IF G#=L$ THEN GOTO 360
150 PRINT TAB 12;G#;" "
160 IF G#=N$ THEN GOTO 350
170 LET W#=N$
180 FOR I=1 TO X
190 IF G$(I)<>W$(I) THEN GOTO 2
300
200 LET W$(I)="P"
210 PRINT "P";
220 LET G$(I)="/"
230 NEXT I
240 FOR I=1 TO X
250 IF W$(I)="P" THEN GOTO 320
260 FOR J=1 TO X
270 IF I=J OR W$(I)<>G$(J) THEN
GOTO 310
280 LET G$(J)="?"
290 PRINT "B";
300 GOTO 320
310 NEXT J
320 NEXT I
330 PRINT
340 GOTO 110
350 PRINT "PARABENS"
360 PRINT "O NUMERO ERA ";N$
370 PRINT AT 21,0;"DESEJA JOGAR
NOVAMENTE ? (S/N)"
380 IF INKEY$="S" THEN GOTO 410
390 IF INKEY$="N" THEN GOTO 440
400 GOTO 370
410 SLOW
420 CLS
430 GOTO 10
440 CLS
450 PRINT AT 10,13;"F I M"
460 STOP
470 SAVE "SENHA"
480 RUN
490 STOP

```

anêndice R

Neste apêndice nós iremos estudar as linhas 590 a 700 do programa PRÉDIOS, também iremos estudar como fazer as devidas modificações nestas linhas. Para facilitar o estudo das linhas vamos desmembrar a linha 610 em várias outras linhas. As novas linhas ficarão assim:

```

601 PRINT AT 21,0;B$
602 PRINT AT 20,0;A$
603 PRINT AT 21,30;" "
604 PRINT AT 13,2;B$( TO 28)
605 PRINT AT 12,2;A$( TO 28)
606 PRINT AT 11,11;B$( TO 8)
607 PRINT AT 17,11;A$( TO 8)
608 PRINT AT 8,7;B$( TO 23)
609 PRINT AT 7,7;A$( TO 23)
610 PRINT AT 5,19;B$( TO 11)
611 PRINT AT 4,19;A$( TO 11)

```

EXPLICAÇÃO

- linha 590 - Iguale a variável A\$ aos pregos martelados (30 pontos).
- linha 600 - Iguale a variável B\$ à plataforma (30 espaços inversos).
- linha 601 - Imprime na posição (0, 21) a variável B\$ inteira.
- linha 602 - Imprime na posição (0, 20) a variável A\$ inteira.
- linha 603 - Imprime na posição (30, 21) dois espaços inversos.
- linha 604 - Imprime na posição (2, 13) a variável B\$, só que agora ela não é impressa inteira, só é impressa até o vigésimo oitavo caractere.
- linha 605 - Imprime na posição (2, 12) a variável A\$ até o vigésimo oitavo caractere.
- linha 606 - Imprime na posição (11, 18) a variável B\$ até o oitavo caractere.
- linha 607 - Imprime na posição (11, 17) a variável A\$ até o oitavo caractere.
- linha 608 - Imprime na posição (7, 8) a variável B\$ até o vigésimo terceiro caractere.
- linha 609 - Imprime na posição (7, 7) a variável A\$ até o vigésimo terceiro caractere.

linha 610 - Imprime na posição (19, 5) a variável B\$ até o décimo primeiro caractere.

linha 610 - Imprime na posição (19, 4) a variável A\$ até o décimo primeiro caractere.

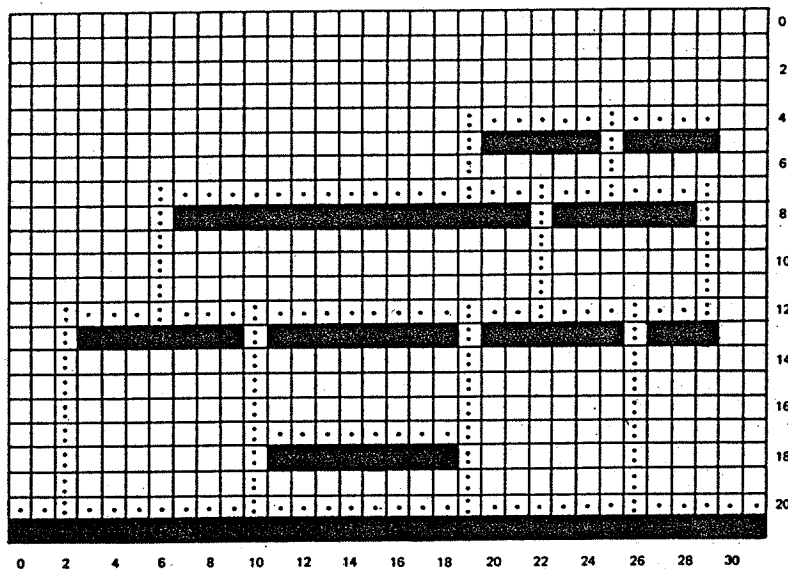
As linhas 620 até a 700 são LOOPS sucessivos para imprimir as escadas.

linha 710 - Passa o computador para o modo SLOW.

linha 720 - Retorna ao programa principal.

Para você modificar o prédio copie uma das páginas do apêndice C e desenhe o seu prédio. Lembre-se que as duas últimas colunas são reservadas para o elevador.

A figura do prédio original é a que segue:



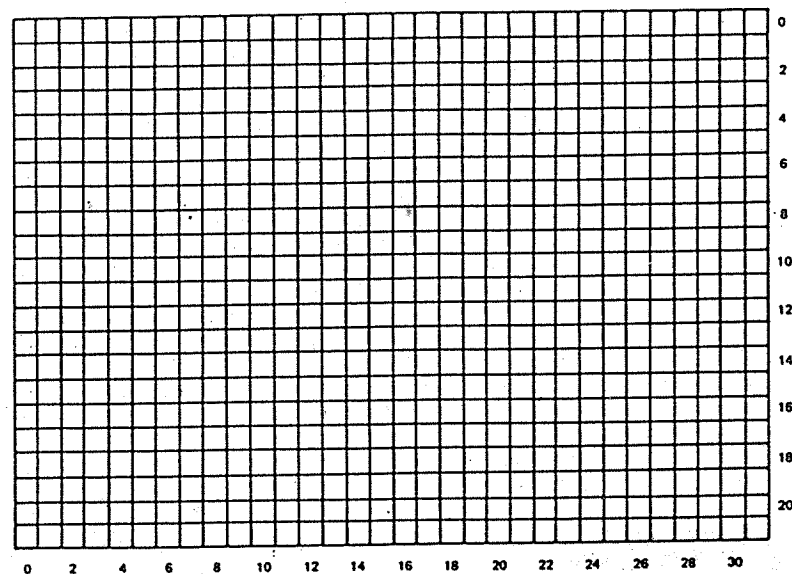
Compare a figura 1 com a explicação das linhas 600 a 700 e modifique o prédio para que ele fique como você quer.

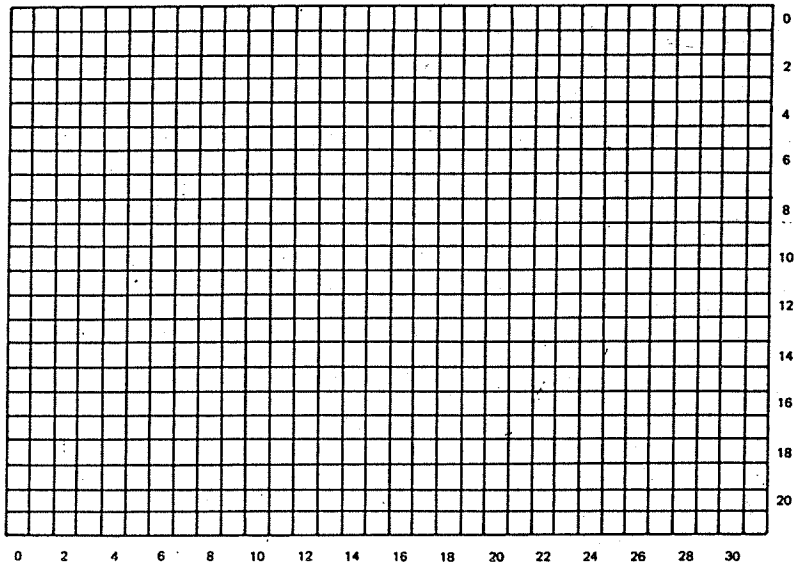
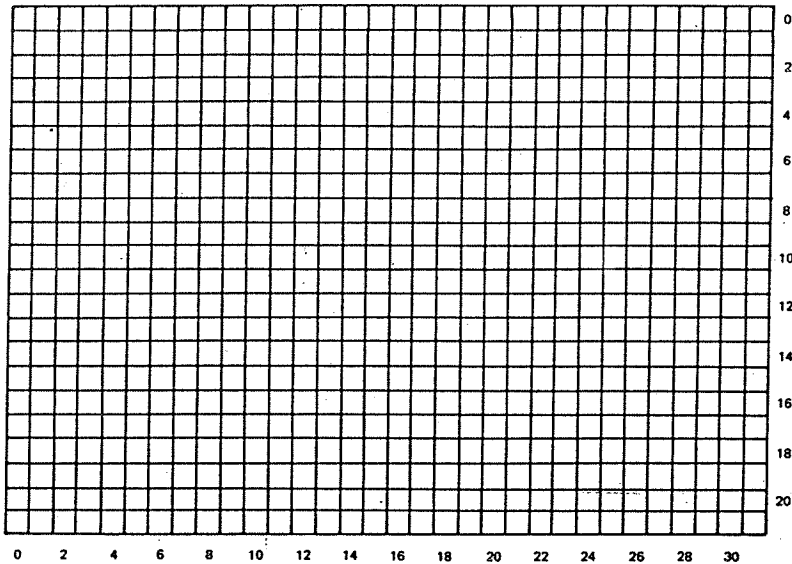
Logo após a modificação do prédio conte o número de pregos que existem para ser arrancados e subtraia 1 e coloque o valor encontrado na linha 360, ela irá ficar assim:

```
360 IF DE = valor encontrado THEN GOTO 540
```

apêndice C

Se você quiser alterar o display de algum jogo, deverá relocar algumas posições de PRINT. Para isto estamos publicando algumas cópias da tela de maneira que você possa se organizar.





apêndice D

Neste apêndice são fornecidos os códigos decimais do TK com os seus respectivos códigos em hexadecimal e seus caracteres, que servirão para o leitor quando for modificar algum programa.

0	00		15	10	(
1	01	#	17	11)
2	02	#	18	12	>
3	03		19	13	<
4	04	#	20	14	=
5	05		21	15	+
6	06	#	22	16	-
7	07		23	17	*
8	08		24	18	/
9	09		25	19	;
10	0A		26	1A	,
11	0B	..	27	1B	.
12	0C		28	1C	@
13	0D	#	29	1D	1
14	0E	:	30	1E	2
15	0F	?	31	1F	3

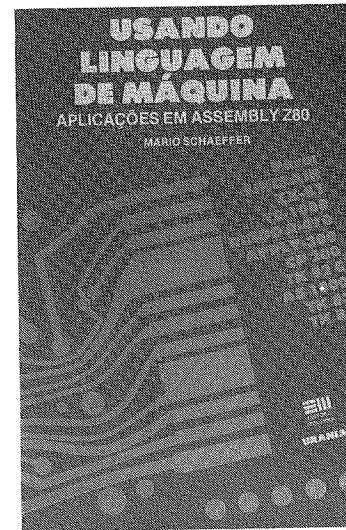
32	20	4	55	37	R
33	21	5	56	38	S
34	22	6	57	39	T
35	23	7	58	3A	U
36	24	8	59	3B	V
37	25	9	60	3C	W
38	26	A	61	3D	X
39	27	B	62	3E	Y
40	28	C	63	3F	Z
41	29	D	64	40	RND
42	2A	E	65	41	INKEY\$
43	2B	F	66	42	PI
44	2C	G	67	43	?
45	2D	H	68	44	?
46	2E	I	69	45	?
47	2F	J	70	46	?
48	30	K	71	47	?
49	31	L	72	48	?
50	32	M	73	49	?
51	33	N	74	4A	?
52	34	O	75	4B	?
53	35	P	76	4C	?
54	36	Q	77	4D	?

78	4E	?	101	65	?
79	4F	?	102	66	?
80	50	?	103	67	?
81	51	?	104	68	?
82	52	?	105	69	?
83	53	?	106	6A	?
84	54	?	107	6B	?
85	55	?	108	6C	?
86	56	?	109	6D	?
87	57	?	110	6E	?
88	58	?	111	6F	?
89	59	?	112	70	?
90	5A	?	113	71	?
91	5B	?	114	72	?
92	5C	?	115	73	?
93	5D	?	116	74	?
94	5E	?	117	75	?
95	5F	?	118	76	?
96	60	?	119	77	?
97	61	?	120	78	?
98	62	?	121	79	?
99	63	?	122	7A	?
100	64	?	123	7B	?

124	7C	?	☐
125	7D	?	☐
126	7E	?	☐
127	7F	?	☐
128	80	☐	☐
129	81	☐	☐
130	82	☐	☐
131	83	☐	☐
132	84	☐	☐
133	85	☐	☐
134	86	☐	☐
135	87	☐	☐
136	88	☐	☐
137	89	☐	☐
138	8A	☐	☐
139	8B	☐	☐
140	8C	☐	☐
141	8D	☐	☐
142	8E	☐	☐
143	8F	☐	☐
144	90	☐	☐
145	91	☐	☐
146	92	☐	☐
147	93	☐	☐
148	94	☐	☐
149	95	☐	☐
150	96	☐	☐
151	97	☐	☐
152	98	☐	☐
153	99	☐	☐
154	9A	☐	☐
155	9B	☐	☐
156	9C	☐	☐
157	9D	☐	☐
158	9E	☐	☐
159	9F	☐	☐
160	A0	☐	☐
161	A1	☐	☐
162	A2	☐	☐
163	A3	☐	☐
164	A4	☐	☐
165	A5	☐	☐
166	A6	☐	☐
167	A7	☐	☐
168	A8	☐	☐
169	A9	☐	☐

170	AA	☐	193	C1	AT
171	AB	☐	194	C2	TAB
172	AC	☐	195	C3	?
173	AD	☐	196	C4	CODE
174	AE	☐	197	C5	VAL
175	AF	☐	198	C6	LEN
176	B0	☐	199	C7	SIN
177	B1	☐	200	C8	COS
178	B2	☐	201	C9	TAN
179	B3	☐	202	CA	ASN
180	B4	☐	203	CB	ACS
181	B5	☐	204	CC	ATN
182	B6	☐	205	CD	LN
183	B7	☐	206	CE	EXP
184	B8	☐	207	CF	INT
185	B9	☐	208	D0	SQR
186	BA	☐	209	D1	SGN
187	BB	☐	210	D2	ABS
188	BC	☐	211	D3	PEEK
189	BD	☐	212	D4	USR
190	BE	☐	213	D5	STR\$
191	BF	☐	214	D6	CHR\$
192	C0	"	215	D7	NOT

216	D8	**	236	EC	GOTO
217	D9	OR	237	ED	GOSUB
218	DA	AND	238	EE	INPUT
219	DB	(=	239	EF	LOAD
220	DC	>=	240	FO	LIST
221	DD	<>	241	F1	LET
222	DE	THEN	242	F2	PAUSE
223	DF	TO	243	F3	NEXT
224	E0	STEP	244	F4	POKE
225	E1	LPRINT	245	F5	PRINT
226	E2	LLIST	246	F6	PLOT
227	E3	STOP	247	F7	RUN
228	E4	SLOW	248	F8	SAVE
229	E5	FAST	249	F9	RAND
230	E6	NEW	250	FA	IF
231	E7	SCROLL	251	FB	CLS
232	E8	CONT	252	FC	UNPLOT
233	E9	DIM	253	FD	CLEAR
234	EA	REM	254	FE	RETURN
235	EB	FOR	255	FF	COPY



**USANDO
LINGUAGEM DE MÁQUINA
APLICAÇÕES EM ASSEMBLY Z80
MÁRIO SCHAEFFER**

Uma obra original, realmente didática, para aprendizado e consulta.

Instruções para o uso de linguagem de máquina em computadores compatíveis com Sinclair (RINGO, ZX-81, TS-1000, NEZ 8000, TK-82C, TK-85, TK-83).

Como usar as sub-rotinas da ROM, inclusive para cálculos científicos.

Mário Schaeffer mostra com muita inteligência e didática, como fazer verdadeiros milagres de programação utilizando Linguagem de Máquina, fornecendo muitíssimos exemplos de aplicação e brindando o leitor com uma série de programas úteis e divertidos.

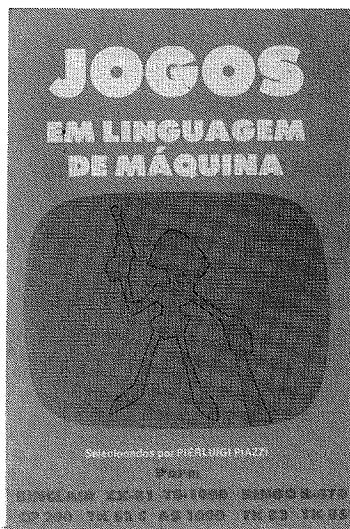


**DISSECANDO JOGOS
em BASIC TK comentado linha por linha
CARLOS EDUARDO ROCHA SALVATO**

Os sete jogos contidos neste volume tem uma característica única: o leitor participa do processo de criação acompanhando a montagem linha por linha.

Modificações e aperfeiçoamentos são propostos e resolvidos com detalhes. Ao terminar este livro o leitor além de ter sete jogos geniais, terá adquirido a habilidade de programar com criatividade.

Uma obra didática indispensável ao usuário de um micro da linha SINCLAIR: Ringo R-470, CP-200, TK-82/83/85, AS-1000, SINCLAIR ZX-81 ou TS-1000.



JOGOS EM LINGUAGEM DE MÁQUINA - VOL I

Selecionados por PIERLUIGI PIAZZI

Use suas habilidades, sua inteligência e seus reflexos: viva os jogos que você mesmo digitou.

JOGOS EM LINGUAGEM DE MÁQUINA é um livro fascinante, rico em explicações do qual você extrairá uma quantidade enorme de programas.

A digitação dos programas não exige conhecimento de linguagem de máquina.

JOGOS EM LINGUAGEM DE MÁQUINA - VOL. II

Selecionados por PIERLUIGI PIAZZI

Para quem já se deliciou com o Volume I desta coleção, mais programas e jogos interessantíssimos e fascinantes:

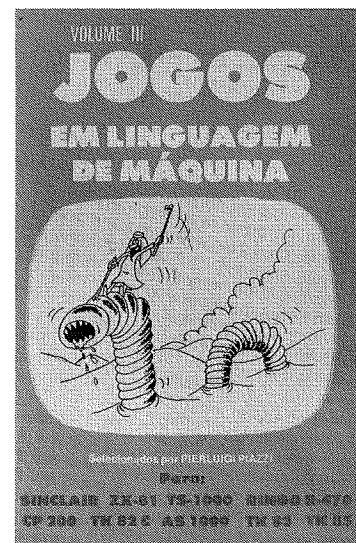
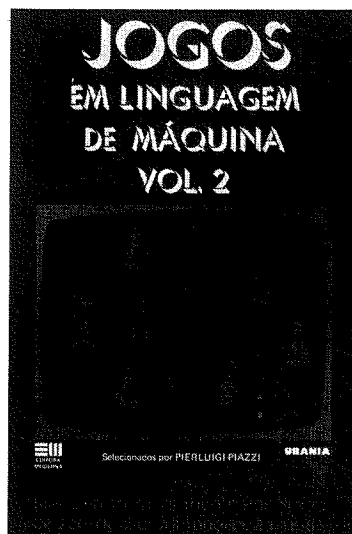
JOGOS DE AÇÃO:

- WARDOZ
- CICLO TRON
- PAC MAN
- CORRIDA DO OURO
- TÚNEL

JOGOS INTELIGENTES: Criatividade e um magnífico xadrez, com instruções detalhadas e informações curiosas sobre este jogo tão antigo e tão atual.

Todos os jogos em ASSEMBLY Z-80, de execução rápida. Uma obra de lazer e de consulta para usuários de micro-computadores compatíveis com SINCLAIR ZX-81: TK-82, RINGO, R-470, CP-200, TK-83, AS-1000 e TK-85.

Pelo preço de uma fita, uma quantidade incrível de programas geniais!



JOGOS EM LINGUAGEM DE MÁQUINA - VOL. III

Selecionados por PIERLUIGI PIAZZI

Continuando a série, mais jogos inéditos:

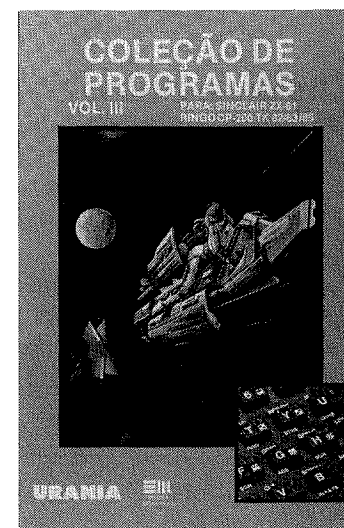
- INVASÃO
- I.R.A.
- VERMES DE AREIA (Duna)
- BASQUETE
- FROGGIE

e muitos outros.

Desenvolva o lado direito de seu cérebro (visão espacial) jogando o fascinante DÉDALO.

A digitação não exige conhecimento de Linguagem de Máquina.

Jogos em ASSEMBLY Z-80 para computadores compatíveis com SINCLAIR ZX-81 (TK-82/83/85, CP-200, TS-1000, etc.).

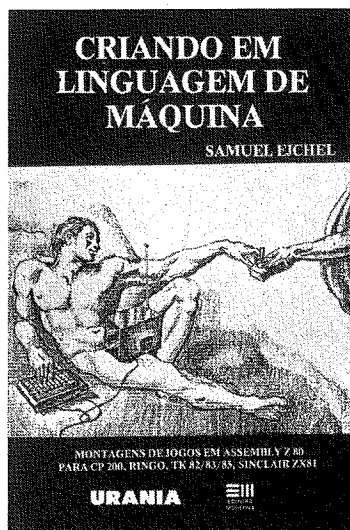


COLEÇÃO DE PROGRAMAS VOLUME III

RICARDO DE FAYETTI SIQUEIRA

Aos iniciantes no uso de um micro-computador pessoal, nada melhor, para o aprendizado, do que a digitação de programas e sua posterior análise e adaptação. Esta coleção pretende fornecer subsídios para o aprendizado e o lazer do usuário, apresentando tanto programas de jogos, quanto programas didáticos na área de matemática, da física e da química.

O Volume III da COLEÇÃO DE PROGRAMAS é portanto indicado para estudantes, adolescentes e iniciantes na fascinante técnica (e arte) de programar um computador pessoal.



CRIANDO EM LINGUAGEM DE MÁQUINA SAMUEL EICHEL

Não basta se conhecer as instruções do ASSEMBLY para se criar um programa interessante: existem truques e "dicas" indispensáveis para o bom programador.

- Gerenciando a tela
- Utilizando o teclado
- Usando o Joystick
- Gerando movimentos múltiplos
- Criando opções
- Implementando "requintes"

Em cada capítulo um jogo original, totalmente explicado, ilustrando os truques de programação utilizados.

Uma obra indispensáveis para quem programa em micros da linha SINCLAIR: CP-200 – RINGO – TK-82/83/85 – AS-1000.

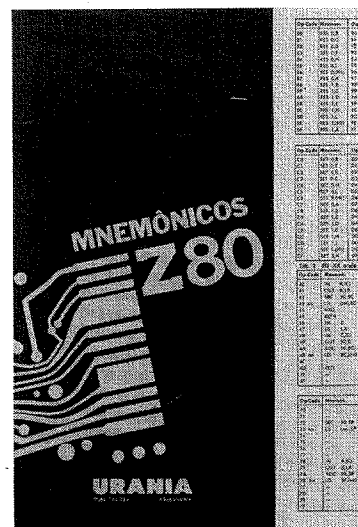
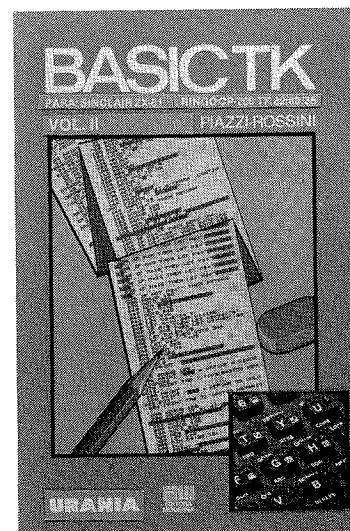


TABELA DE MNEMÔNICOS Z80

Tabela de cartolina plastificada para consulta rápida das instruções do Z80, seus códigos hexadecimais e abreviações mnemônicas. Indispensável para quem programa em linguagem de máquina.

Para computadores da linha SINCLAIR ZX81 (TK-82/83/85, CP-200, Ringo, AS-1000) e TRS80 (CP-300/500, JR SYSDATA, D-8000/8001/8002, DGT-100/101).



BASIC TK - VOL. II PIERLUIGI PIAZZI/FLAVIO ROSSINI

A continuação de um dos maiores sucessos editoriais em informática.

Aprofundamento do BASIC com estrutura de programação.

Truques utilizando o sistema operacional dos micros da linha SINCLAIR (ZX-81, TK-82/83/85, CP-200, RINGO).

Uma obra elaborada com os mesmos cuidados didáticos do Volume I.

LANÇAMENTO!!!



EVOLUINDO NO BASIC TK BERNARDO STEIN

Uma coletânea de programas em BASIC explicados detalhadamente, que transforma o micro em:

- O DESENHISTA
- O SENTINELA
- O PROFESSOR

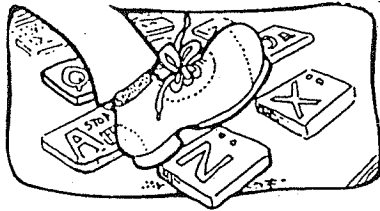
e mais uma série de interessantíssimas funções.

Um livro didático escrito num estilo simples e bem humorado para usuários de TK-82/83/85, CP-200, RINGO e ZX-81.

Querendo adquirir um destes livros escreva para:
ALEPH

PUBLICAÇÕES E ASSESSORIA PEDAGÓGICA LTDA.
Av. Brig. Faria Lima, 1451 - conj. 31
01451 - São Paulo - SP - Tel.: (011) 813-4555

O IMPORTANTE É O PRIMEIRO PASSO



CURSOS

BASIC I e II (crianças e adultos)
JOGOS (crianças e adolescentes)

Coordenação didática:
prof. PIERLUIGI PIAZZI

NÚCLEO DE ORIENTAÇÃO DE ESTUDOS
Av. Brigadeiro Faria Lima, 1451 - conj. 31 -| São Paulo
Informações: (011) 813-4555

LIBERTE O GENIO DO SEU MICRO



CURSO DE Z80

CURSO DE ASSEMBLY

O seu micro de lógica SINCLAIR (RINGO, CP-200, TK82/83/85, AS 1000, etc.) bem como os compatíveis com TRS-80, têm um microprocessador Z-80.

Aprendendo Linguagem de Máquina (ASSEMBLY Z 80) você poderá comandar diretamente o microprocessador

estruturando programas muitas vezes mais rápidos do que em BASIC e gastando muito menos memória.

Você poderá aprender ASSEMBLY como fez o FLAVIO ROSSINI (autor do LINGUAGEM DE MÁQUINA PARA O TK - Ed. MODERNA) assistindo às aulas do professor FRANCISCO A.S. DE OLIVEIRA.

Os cursos LM I (40 horas) e LM II (40 horas) são no NÚCLEO DE ORIENTAÇÃO DE ESTUDOS - Av. Brig. Faria Lima, 1451 conj. 31, em vários horários, inclusive aos sábados. Para informações ligue para (011) 813-4555.



impresso na
planimpress gráfica e editora
rua anhaia, 247 - s.p.